Informatik
Johannes Kepler Universität

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Gesture-Based Browsing of Mathematics

Master Thesis

Author:	Shereen Abd El Halim El Bedewy
Supervisors:	Prof.Dr.Dr.h.c.mult.Bruno Buchberger
	DI Dr.Wolfgang Windsteiger
	a.Univ.-Prof. Dr. Klaus Miesenberger
	Dipl.-Ing. Bernhard Stöger
Submission Date:	12 July, 2013

Informatik
Johannes Kepler Universität

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Gesture-Based Browsing of Mathematics

Master Thesis

Author:	Shereen Abd El Halim El Bedewy
Supervisors:	Prof.Dr.Dr.h.c.mult.Bruno Buchberger
	DI Dr.Wolfgang Windsteiger
	a.Univ.-Prof. Dr. Klaus Miesenberger
	Dipl.-Ing. Bernhard Stöger
Submission Date:	12 July, 2013

Ich erkläre an Eides statt, das ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw die wörtlich oder sinngemäss entnommenen Stellen als solche kenntlich gemacht habe.

This is to certify that:

(i) The thesis comprises only my original work toward the Master Degree

(ii) Due acknowledgment has been made in the text to all other material used

---

Shereen Abd El Halim El Bedewy
12 July, 2013

Acknowledgments

I would like to express my sincere gratitude to Dipl.-Ing.Bernhard Stöger for his help and support through all my work.

My sincere appreciation will never be enough to DI Dr.Wolfgang Windsteiger who helped me in understanding Mathematica and achieving the implementation phase. To Prof.Dr.Klaus Miesenberger for supporting my thesis and giving me a place in the Integriert Studieren Institute at the JKU.

Writing now my master thesis degree is honored by my great Professor Prof.Dr.Dr.h.c.mult.Bruno Buchberger who gave me the chance to study in Austria, get a wonderful thesis project and achieve a dream I was always looking for.

My appreciation goes from my heart to all of you, giving me support, guidance, encouragement and enriching me with knowledge throughout my study.

On the personal level, my family whom I will never be able to pay them back the love and support they always provide me.
My gift from God, my parents were always the greatest motive of my life. Looking at the success they always achieve makes me want to follow their footsteps. My mum Prof.Dr.Hala Ezzat, I thank her for her precious time in making me who I am. My dad Dipl.Eng.Abd El Halim Elbedewy following his foot steps into Informatics and for his endless advices, I learned from him a lesson of my life which is never to stop dreaming and never give up, because one day you will achieve it all.
My sister Hanzada and my brother Aly for being such good friends and supportive, loving people.

To Mohamed the man who is always standing by my side, pushing me forward supporting me with his love and care. My gratitude can never be enough.

I can't forget all those who added knowledge to me, my professors I thank you all. At last I want to express my special thanks to all those who achieved many things in this field, without them I wouldn't have started.

"Success as I see it, is a result, not a goal" - Gustave Flaubert

Abstract

This paper is introducing a new concept which is combining gestures and speech in applications to help people navigate through Mathematica notebooks. The application can be used by all people, but it is targeting visually impaired people specifically.
Starting from the background of gestures, speech and all the offered tools to help blind people till we reach the challenges that we met in our approach is all covered in this thesis. On the basis of the $1 gesture recognition algorithm our definition of gestures started and advanced to include the enhancements that took place in the $1 gesture recognition itself.
Introducing fresh new ideas as a future research. Showing the connection of the phrase "Application for visually impaired people" to many other scientific fields and specifically to mathematics. Moreover mentioning the importance of mathematics and its different software tools that were made to help visually impaired people, is all covered in this content.

Keywords: Mathematics, Mathematica Notebooks, Touch Gestures, Visually Impaired People, Speech Output, Gesture Recognizers.

Contents

Acknowledgments **V**

1 Introduction **1**
1.1 Mathematics in our life . . . 1
1.2 Math Linearization . . . 3
1.3 Math is Blind !! . . . 6
1.4 The Project's Goal . . . 7

2 State Of The Art **8**
2.1 How do Visually Impaired People use Computers ? . . . 8
2.2 Computer Based Math and Computer Algebra . . . 9
2.2.1 Matlab . . . 9
2.2.2 Mathematica . . . 9
2.2.3 R . . . 9
2.2.4 LaTeX . . . 9
2.3 Software Tools and Visually Impaired People . . . 10
2.4 Touch Interaction and Audio Technology . . . 13
2.5 Mathematica as a Choice . . . 14

3 Specification and Development of GBM System **16**
3.1 Project Steps . . . 16
3.2 Project Structure . . . 17

4 User Manual **19**
4.1 Organization . . . 19
4.1.1 Modes . . . 19
4.1.2 Button Notation . . . 19
4.1.3 Interaction Commands Gestures . . . 20
4.2 Examples . . . 20
4.2.1 The Higher Level "Notebook" . . . 21
4.2.2 The Lower Level " Cell" . . . 24

5 Implementation **26**
5.1 Software Integration . . . 26
5.2 Code Overview . . . 27
5.2.1 $1 Algorithm . . . 27
5.2.2 The GBM Coding . . . 29

5.2.2.1 Step One: Resampling . . . 30
5.2.2.2 Step Two: Rotation based on "Indicative Angle" . . . 31
5.2.2.3 Step Three: Scale and Translate . . . 33
5.2.2.4 Step Four: Optimal Angle and Best Score . . . 33
5.2.3 Limitations of the $1 Algorithm . . . 34
5.2.4 Enhancements on $1 Algorithm . . . 39
5.3 The NoteBook Interactions . . . 42
5.3.1 The Higher Level "Notebook Level" . . . 42
5.3.1.1 Pattern Matching . . . 43
5.3.1.2 Four Pattern Matching Categories . . . 44
5.3.2 Lower Level "Cell Level" . . . 45
5.3.2.1 The Cases . . . 46

6 Evaluation **48**
6.1 The Evaluation Methodology . . . 48
6.2 Users Feedback . . . 49
6.2.1 Challenges . . . 49

7 Future Work **52**
7.1 Insights Into the Future . . . 52
7.2 Future Work Enhancements to the GBM System . . . 53
7.2.1 Functionalities . . . 53
7.2.2 Enhancing the GBM System . . . 55

8 Conclusion **57**
8.1 The Sum Up . . . 57
8.2 The Glory of Blind People Capabilities . . . 59

Appendix **60**

A Implementation Code **61**
$1 and GBM Algorithm . . . 61

B Lists **65**
List of Abbreviations . . . 65
List of Figures . . . 67

References **70**

Chapter 1

Introduction

> "Everything has its wonders, even darkness and silence, and I learn, whatever state I may be in, therein to be content." Helen Keller [1]

1.1 Mathematics in our life

Mathematics plays an important role not only in our educational life but also in our daily life. As a good start let us know what mathematics is, the definition of mathematics from Britannica concise encyclopedia is:

"Science of structure, order, and relation that has evolved from counting, measuring and describing the shapes of objects. It deals with logical reasoning and quantitative calculations." [1]

On the other hand one can notice that the literal meaning of mathematics is much simpler than its definition, it is simply "Things which can be counted". Counting is considered a very important method in our daily life, as we count our age, we count the days, months and even years.When we dial numbers on the phone we use counting, when we travel to some place we count the distances, when we play sports we count the score of the winning team, to measure cloth size all this require mathematics and counting.

From the basic life activities to the advanced ones, we need mathematics, we may sometimes use formulas and form equations in our daily life while we are not even noticing. For example, in cooking the ingredients are defined in specific portions and quantities and if any portion increased or decreased the output is not always the same e.g. For a Chocolate cake one needs: 5 eggs, 3/4 cup of sugar, 1/2 cup of vegetable oil..etc. In medicine, how many tablets should the patient take. In Banks when people save their money and open accounts. Therefore, on a basic level one needs to be able to count, multiply, subtract and divide to interact with the surrounding environment.[2]

[1] Helen Adams Keller (June 27, 1880 June 1, 1968) was an American Author political activist, and lecturer. She was the first deaf-blind person to earn a Bachelor of Arts degree.

Math is considered universal as since ages human beings discovered math concepts, they didn't invent them. Even one can notice that math language is numbers not Japanese, English ,German or even Russian. If we are well educated in this language of numbers, it can help us make important decisions in our lives and perform everyday tasks more efficiently. Math can help us shop wisely, remodel a home within a budget, buy the right insurance for ourselves, or even bet on the horse with the best chance of winning in a race. All these examples were to reflect the usage of math in our daily life. [3]

> "Education should be started with mathematics. For it forms well designed brains that are able to reason right. It is even admitted that those who have studied mathematics during their childhood should be trusted, for they have acquired solid bases for arguing which become to them a sort of second nature".- Ibn Khaldun[2]

Mathematical content is an important aspect in any scientific field. Consider its role in, physics, economy, chemistry, biology, and computer sciences, mathematics is important in people's lives, therefore they try to make its content accessible in the easiest way.

For example, by assisting the mathematical content representation by visual aids as drawings and notes which illustrate the main content as geometrical figures (lines, curves), graphical notes (strokes, underlines, highlighting some parts of the material or links between different terms).

Relying on the fact that mathematical functions or equations is usually displayed in two dimensions, helps the reader to convey the general structure of the formula, and it becomes easier to understand its semantics. [4]
As it is clear in the formula below in Figure 1.1, the 2D representation of this mathematical formula is easy to interpret, as one can easily detect the fraction (Denominator and Numerator) as well as the square root and what variables does it enclose, one can also distinguish the difference of the left hand side and the right hand side of the equation easily which is separated by the = sign.

$$L_1 = \frac{L_0}{\sqrt{1-\frac{v^2}{c^2}}}$$

Figure 1.1: Mathematical formula displayed in 2D

[2]Ibn Khaldun, al-Muqaddima (born in 1332, Tunis), historian, sociologist, philosopher. Strongest personalities of Arabo-Muslim culture in the period of its deline.

But having a look in Figure 1.2 on the same formula in a linear form, one could realize the difference. It is longer and complicated, a normal math student would have hard time to distinguish the parentheses and trace each one, to know when they open and when they close. The fraction part is not obvious and also the square root. This illustrates that mathematical formula are better represented in 2D.

$$L_1 = L_0/Sqrt(1 - v\hat{}2/c\hat{}2)$$

Figure 1.2: Mathematical formula displayed in linear form

On the other hand, people also considered the Software in making math easier, as we can see in Software like Matlab, Mathematica and R. Which help people perform the mathematical tasks and its representation in an advanced easy way.

In both cases either considering paper-pen or software in accessing math, one can notice the fact that the sight plays an important factor in providing a global perception and understanding of the mathematical expressions and structures. The reader can perceive the structure of an expression, localize its elements, and access each of these elements by a deep look into any mathematical expression. After analyzing the expression a tree representation of this expression is built mentally step by step before its leaves are even read. The glory of math in building a link between the visual representation and the brain is revealed.

After considering the importance of sight in accessing the mathematical content, let us have a deep look on the case of its absence. Visually impaired people face great challenge in dealing with very complex mathematical content. Therefore they have to rely on other communication channels that are available to convey the mathematical contents as audio and tactile. These reasons make mathematical concepts much more difficult for blind individuals than for sighted ones.[5]

1.2 Math Linearization

Considering other ways of communication for visually impaired people, speech can make great use of audio and Braille can make use of tactile.
According to Braille Authority of North America (1944), the Braille definition is a tactile(touch) writing and reading system used by the blind and the visually impaired people. The main component in the Braille is the cells which contain dots that are arranged in certain order. For example the cell has three dots as its height and two dots as its width. Though one thinks all characters and symbols can be formed with Braille, but the fact is that only 63 different characters are achieved using the Braille. The Braille can be found in various places as in books, on menus, signs, elevator buttons, and even on currency. Braille made it possible for visually impaired people to read and write.[6]

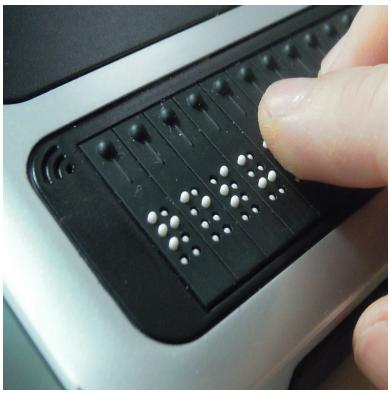


Figure 1.3: The Braille Display
[7]

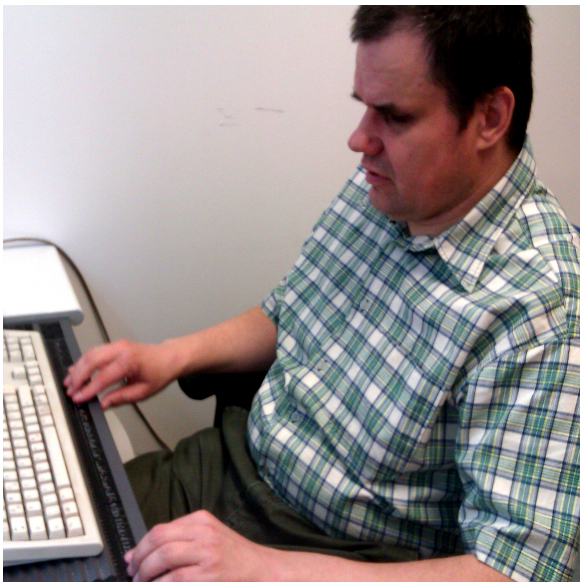


Figure 1.4: Visually impaired users using the Braille display
[8]

But both Speech and Braille are intrinsically linear, which means that formulas need to be linearized and in most cases this linearization generates a much longer representation and expressions cannot be presented in two dimensions, which is more difficult to understand by blind individuals than the graphical presentation for the sighted individuals.[9]

The linearization of any Mathematical expression displayed in two dimensions has two effects, the linear version is bigger (It has more signs) and the immediate access to the structure of the mathematical expression is lost. Therefore the mental representation must be built during the actual reading of the expression, including the detail of each term. And the tree representation will be clear after the expression is finished and after the leaves are reached.[4]
This makes it difficult to achieve overall understanding and to directly access specific sub expressions. Therefore, reading a Braille representation of a mathematical expression is a sequential process. The overall meaning of an expression is understood by finding common relations among

simple sub expressions. The possibility to perceive further exploration and to access specific parts directly can be very hard when linearization takes place.

On the other hand reading and understanding a mathematical expression through speech output or Braille, lacks the solid representation of the structure.[10]
In case of speech and auditory representation (such as audio recordings) of any written content, the traditional reading process is reversed. As the normal reading process is an active-passive process, it includes two partners, the first partner is the sighted person the active participant while the book they are reading is the second passive partner. As audio representation of the written content takes place, the reader has limited freedom and the only way he/she can be active is to move forward or backward in a time-line in the whole reading process. In this particular case the book takes the place of the first partner and becomes the active one, and the reader becomes a passive listener with limited freedom. This issue can be very difficult when the content holds technical material as for instance statistics, mathematical or scientific content. Because the way in which people read scientific or technical documents should be quite different from how novels and literature is read.[11]

Visually impaired people when they read equations they would be in a very difficult situation in performing mathematics if they only heard equations. The main thing in the presentation of mathematical material is reflecting the structure of the content, this can be done by grouping of an expression using pauses and alterations in the speaking rate. Sometimes an initial scan of the equation is recommended, because this allows the readers to imagine the representation of the equation structure, followed by a detailed imagination when moving from an element to another filling in mentally the empty spaces. This would overcome the difficulty blind people face when they hear scientific content with equations represented in a strictly linear form, as occurs with many browsers for blind equation readers. This issue is handled in our Gesture Browsing of Mathematics (GBM) system, as the user usually takes an overview of the content and the structure of the whole notebook and then moves step by step on lower levels of the presentation of the Mathematica notebooks.[12] [11]

Regardless of the difficulty Speech and Braille present, mathematical expressions can still be understood effectively and efficiently but only when the reader can rely on operations to rapidly access specific parts of the expression and to extract the overall structure and this is what we are eager to achieve in this paper.

1.3 Math is Blind !!

This paper and its state of art try to solve the problems blind students tend to face when they are learning mathematics. Blind students face multiple challenges in math problem solving, including gaining access to the problem information, mapping the problem information to the appropriate representation, and providing the resulting answer to each part. [13]
The current state of the paper was able to solve the following challenges:

1. Gaining access to the problem information.
2. Mapping the problem information to the appropriate representation by the speech presentation.

The research in the field of offering mathematics to blind people revealed the nature of math, which is math is blind. The actual mathematical activity reasoning is happening in the brain without involving senses (eyes) and neither actors (hands). Math is blind because regardless of the way we input math whether by visual, auditory or tactile aid it is always perceived and presented to the brain not to the eyes, the ears or even to the hands. This concept is inspired from Prof.Bruchberger's talk in the Keynotes at the International Conference on Computers Helping People with Special Needs 2012. In this conference he presented the idea of why math is blind in its structure.[14] [15]

Keeping into consideration the great use of the tactile communication channel to the visually impaired people, the Braille is considered one of the best examples that help blind people use the touch as a source of communicating with computers, laptops and access the Internet. But unfortunately there are some disadvantages that come with the Braille which is the linearization of the mathematical content which makes it longer and more difficult to imagine the structure of the expressions as mentioned before.
Another problem arose with the Braille is what so called "Braille representations" of equations for example in the mathematical content, there are a large number of such representations in the world, and unfortunately, very few are the same. Moreover blind mathematicians depend on their own Braille representations of equations, that they trained on and know how to interact with. Thus, a mathematician in the U.S. using Nemeth Braille cannot easily send his/her work to a colleague in Germany, where Marburg Braille is the standard.[16]

1.4 The Project's Goal

Giving a brief introduction of Mathematics in our life and covering some issues about blind people interaction with Mathematics. Here comes the part which reflects to our readers the points, topics and achievements covered in this theses. The users of this application could be blind, partially sighted and fully sighted people, therefore we are going to generalize and refer to the three categories as users. The approach introduced in this paper relies on tactile "gestures" as an input mode from the user and audio as an output mode to the user.

The main aim of the application is to build an interface to accept gestures from the users and according to these gestures, the user will be able to navigate through the Mathematica notebooks.

The Mathematica notebooks could hold any type of cells (text, proofs, mathematical expressions and equations). A combined sets of cells is simply the structure of the Mathematica notebook, it includes all the usual features of a word-processing system and other special capabilities. Mathematica notebooks have a great symbolic structure that allows full markup, cascading style sheets, and offers the option of restyling a document. Notebooks can be used efficiently not only for interactive use, but also for export to web and print media.[17]

The application supports navigating through existing notebooks and exploring their content from cell to cell until the whole picture of the notebook is reflected to the brain while preserving the same structure.

Unlike the true meaning of gestures which is the act of moving the limbs or body as an expression of thought or emphasis. We mean by gestures here a special motion by the finger on a touch slate to indicate a shape, based on each shape an action will be performed. The gestures used in this application are simple enough to help the users focus on the main content of the notebook and doesn't require any learning process in advance. One of our main concerns was to minimize the number of gestures used to make the memorization processes easier for the users. Therefore we are sticking to 7-8 gestures at most, each one is responsible for a certain action as we will see in the coming chapters.[18]

The hypothesis question one could ask,

Can blind people use gestures for browsing mathematical content ?

Chapter 2

State Of The Art

2.1 How do Visually Impaired People use Computers ?

This part shows how visually impaired people interact with computers and existing equipments. As we all know the importance of computers and Internet in our daily life, they are also important to handicapped people. Therefore current research is trying to remove or overcome the barrier between people with disabilities and technology accessing tools(Computers, Mobile phones..etc). Therefore we dedicate this section to show how technology made it possible for blind people to use computers, access the Internet and to use smart phones.

Blind people use computers through what is called "Screen Readers". A screen reader is simply a computer application, which runs parallel to other programs its main aim is to inform blind people on what is being displayed on the computer screen. The screen reader has a built in speech synthesizer which speaks the displayed material of the computer screen to the blind users and allows the conversion of the displayed content on the screen to a different format which is readable to the users. Typically we are pointing to the Braille format, which displays the content in the form of a row of dots and pins that pop-up according to the displayed material. The user then can go along the pins with his finger to read what is displayed on the screen, because the Braille display by definition makes the information appear on a Braille line with a certain number of dots and pins, which blind people can read with their fingers.
Another format we are pointed to, is the speech output with the speech synthesizer, this allows the conversion of text to speech directly. It performs this by reading out loud what is displayed on the screen including pop-up boxes, command buttons, menu items and plain text.
On the other hand there is another group of people which are partially sighted, therefore they require other kind of screen readers called "Screen Magnifiers" which magnifies and enhances the displayed content on the screen. [19]

These equipments the Screen Reader, Screen Magnifier, Speech Synthesizer and finally the Braille display are all means to help blind people interact with computers and in our case we would say interact with mathematics on the computer. This is to give the reader an overview on how blind people can read the mathematical notebooks using assisting applications and programs, before we introduce our approach. Our approach in fact doesn't need any assisting applications, but moreover it requires turning off these applications while running the GBM system to avoid any interruption.

2.2 Computer Based Math and Computer Algebra

Mathematics can be accessed through many ways as we discussed before, through paper-pen technique or software based accessing. We introduce the powerful software tools "Matlab, Mathematica, R and LaTeX." and the difference between them, moreover the functionalities they offer to help blind people.

2.2.1 Matlab

Matlab is considered a high-level language, the user can interact with this language to do numerical computations, visualization, or programming. Matlab can help users analyze data, develop their own algorithms and create applications. The built in functions help the user to develop and find solutions faster than normal programming languages as C/C++ or Java. This didn't prevent Matlab from connecting and initiating a link to C/C++ programs which makes the applications more powerful. Bearing in mind that Matlab's original use was for the quick calculation of matrix algebra.[20]

2.2.2 Mathematica

Mathematica is a powerful tool in many ways, Mathematica is a general computer software system and language intended for mathematical applications. It can be used as a numerical and symbolic calculator where you type in questions and Mathematica prints out the answers.
Mathematica can act as a visualization system for functions and data. Users can use the high-level programming language which Mathematica provides to create programs.
Mathematica is also a modeling and data analysis environment and a software platform on which you can run packages built for specific applications like the one we are using in our application.
A way to create interactive documents that mix text, animated graphics and sound with active formulas, as one can find many documents that are written in Mathematica and especially the ones which include mathematical functions.
Moreover Mathematica is a control language for external programs and processes and can link to them easily through the "MathLink" to initiate a link to C/C++ or Java or other programming languages or it can be also an embedded system called from within other programs. [21]

2.2.3 R

R is both a language and an environment, its main aim is to perform mathematical, statistical computations and graphics. One particular strengths of R is its ability to design high quality publications including plots and mathematical symbols and formula it also has its own LaTeX-like documentation format for documents. R as an environment includes effective data handling options, a number of operators for calculation on arrays and matrices, tools and graphical facilities for data analysis, display on-screens or on hard copies, and last but not least a well-developed simple programming language.[22]

2.2.4 LaTeX

Taking into consideration that blind people find a difficulty in accessing mathematical content, because most formats are not compatible with the Braille display. Therefore we can't discard the great use of LaTeX as a software tool for visually impaired people.

LaTeX is a system to write and prepare documents, it is a language which can be used on many editors. Like Java which can be used on Eclipse, JCreator or other editors. The format of the LaTeX documents should be ".tex" these could be created using any editor and some editors were created just for writing LaTeX. LaTeX allows writers to focus on the content they are writing and forget about the presentation and the graphical issues. They just put margins while they are writing like specifying titles, sections, sub sections..etc. And LaTeX controls the graphical presentation and converting the .tex files to neat PDF formats. LaTeX also allows the cross referencing between sections and the automatic bibliographic technique, which eases the citation task for the writers. LaTeX is used for many purposes for example in academic fields to write papers, articles or theses because it has many supported packages for example in writing mathematical formulas and functions which is often needed in the scientific fields. Other purpose which is our main concern is its relation with Braille and what functionalities does it offer.[23]

LaTeX2e Package:
This is the main package which LaTeX provides that is compatible with Braille, as it gives the chance for people who don't know how to use Braille to produce Braille documents on papers. It has two essential python scripts (grade1.py, grade2.py) they are responsible for converting the original text to Braille tags, then LaTeX2e package prints these tags as defined Braille symbols on papers.[24]

2.3 Software Tools and Visually Impaired People

As we have seen in the previous section the mathematical software tools as Matlab, Mathematica and R. These tools can offer help to blind people in reading/writing mathematical functions and formulas. Also the speech command to speak out the written content can be of great use to the visually impaired people. Or by connecting the program to any external device to help navigate through the content like Braille or touch slates, or even by constructing a link between any other programming language as Java, C or C++. All these options can be used to build interfaces to help blind people and ease the mathematical operation. Following some examples on how the software tools can help blind people.

- Matlab Report Generator is a simple example on how blind people can benefit from matlab, it can be used to create end-user documents of many forms including PDF, HTML, XML, and even Microsoft Word, with or without the creation of a LaTeX document.

  As most publishers don't put into consideration the accessibility of the papers, documents or journal papers, some formats are very hard to be read and recognized with the visually impaired people applications. The journal articles can be downloaded as plain HTML, PDF files or graphical images as TIF files and these formats are hardly accessible by blind individuals. Because HTML files don't include graphs, tables or even equations and TIF files are not accessible. Some PDF formats are readable only that they sometimes get mixed with printed pages and scans which make the content incomplete.[25]

- "I would like my experience and success with the use of Mathematica to be an inspiration for other blind individuals who are considering a technical profession." Chuck Strickland

Chuck Strickland is an optical blind physicist working in a top government agency. He uses Mathematica to develop and test algorithms that are used to design optical systems for space telescopes.

> "I found learning the syntax of the Mathematica language and the command structure to be very user friendly and consequently painless." Chuck Strickland.

When Strickland worked with Mathematica, he was able to develop a Mathematica program to help him in his technical writing that converts Braille into text, by inputting Braille into his computer with an electronic notepad that stores the Braille in its internal memory. He was able to edit and review the Braille to adjust it upon his satisfaction. Then the electronic Braille is transferred to the PC where the Mathematica program converts it to text. The text can then be adjusted with a word processor.[26]

He has also written a Mathematica procedures that replace certain character sequences with equations or other graphics. He currently uses Mathematica in the first phase of the optical system design. With Mathematica, Strickland constructs a very basic theoretical model that traces only one or two light rays through the optical system's lenses, mirrors, prisms, and diffraction. [26]

> "Mathematica helps reveal the subtleties of the design". Chuck Strickland

- LaTeX in Practice
" Blind students in mathematics, science, and engineering courses encounter serious challenges as they attempt to translate complex equations from print to Braille and from Braille to print. Mathematician Al Maneki and Alysha Jeans, an electrical engineer working in Virginia, draw upon their experiences as blind professionals as they describe an option that has exciting possibilities." [27]

As we have seen LaTeX is a user friendly tool for sighted and blind people. As blind students can solve an assignment, type it in LaTeX, compile it to any visual format and submitting it as any other student in class. Many screen readers were designed to make LaTeX so flexible to blind users as the MathType software for example which allows blind users receive a printed version with Braille notation from a Word document in cooperation with DBT software from Duxbury Systems or Tiger Software Suite from ViewPlus. Using MathType a blind user writes a LaTeX code on a word document then converts it to mathematical symbols and notations. The document can be printed or converted to Braille with DBT or Tiger Software as we have just mentioned. Moreover LaTeX offers the blind users the facility to browse mathematical resources on the web, as some websites include the LaTeX syntax for the mathematical notations. These tools enlarge the facilities LaTeX offers to blind people and make life easier to them.

- LaTeX2Tri
  This is a new conversion from LaTeX to Tri "Triangle", which allows scientific files as mathematics or physics files to be read and understood to blind people. Most of the file types accessible to blind students as Tex, PDf or Word can be converted to Triangle using the LaTeX2Tri.[28]

  The Triangle language is simply the combination of Rich Text Format (RTF) and Triangle True-Type font. The RTF is a word processor, this processor has defined fonts for the representation of mathematical symbols and operations and the output is spoken by a speech synthesizer. [28]

  WinTriangle: is a software that facilitates to the blind people many tasks as hearing and writing the mathematical functions and performing calculations during writing text. The WinTriangle is so important and useful for blind people as the paper-pen are useful for sighted people. Visually impaired people using this software can directly print the documents to the Braille display, do their assignments in class and read other peoples work and write their own publications. It is useful as it is a link between the blind and the sighted people.[28]

  Taking an overview of the LaTeX, Triangle and the WinTriangle, one can trigger the use of each. The sighted people communicate with each other using LaTeX and blind people communicate with each other using WinTriangle which can be read by sighted people as well. But the missing connection is that between blind people reading the sighted peoples work, therefore the LaTeX to Triangle is being introduced. The use of the LaTeX2Tri can be seen at Harvard University were the conversion of the physical and mathematical content of textbooks, papers, journal articles and problem sets was achieved easily using LaTeX2Tri conversion mechanism.[28]

2.4 Touch Interaction and Audio Technology

Many projects and applications tried to combine tactile and audio to help the visually impaired people. We will have a brief look on these projects that have been carried out since a few years, their main aim is to support the work of blind users, facilitating their understanding and helping them to carry out calculations.

Audio System for Technical Reading (ASTER) system aims to produce accurate renderings of documents marked up in the TEX-family of languages. This work was done by Raman, a blind computer scientist who works in accessibility research as auditory user interfaces and speech interaction. ASTER uses both spoken and non-speech audio to assist in this process. The tool used by the ASTER system to produce the audio output, is a language invented by the author, known as Audio Formatting Language (AFL). The aim of this language is to provide mechanisms to control the multiple ways of the audio presentation, such as speech-based, and non-speech sounds. The traditional mathematics notation all people are used to, uses a set of primitive design notations to achieve complex structures as Raman said. The main aim of ASTER's audio representation of mathematics is to produce a non-visual solution to the complex visual representation of mathematics. [4]

The Maths Genie is a formula browser that facilitates understanding of formulas using voice. It has been designed to express the structure of the Mathematical expression and its contents. It works by rendering the graphical output to be synchronized with the audio. The current version supports English, French and Spanish for speech, and offers facilities to add any local language for which the speech synthesizer is available. [16]

Lambda is a mathematical reading and writing system designed for blind students. The main characteristic of the Lambda project is that it is built on a brand new code. This code is an XML code specifically designed for supporting the Braille. Lambda comes with a dedicated editor, which includes navigation support, input functions (keyboard shortcuts, menus, tool bar). It outputs Lambda Braille, speech synthesis were the mathematical symbols are spoken in a descriptive language, a visual presentation in a linear code were it uses a specific font for each Braille character and finally a graphical rendering of the visual presentation.[9] [29]

Introducing now an editor Infty which allows people with and without visual disabilities to share the same technical content. Infty is a well known project in this field as it provides access to mathematical content specifically. The core module of this project is InftyReader which is a specialist in mathematical documents. It has the ability to extract the mathematical formulas in a document and to distinguish the mathematical structure, characters and symbols from other text. It transforms the mathematical representation into XML format, this format can be further converted to well known easy accessible formats as MathML, LATEX, HTML. This study group added more findings and research projects into this field as InftyEditor which allows people to edit the output of InftyReader. InftyEditor has a handwriting dialog in which users can write formulas. ChattyInfty is a mathematical-document editor which gives the blind people the opportunity to access mathematical and scientific documents with speech output.

InftyEditor working in parallel to display the same content for sighted people. ChattyInfty reads to the blind people the text and the mathematical expressions. Blind people can use the ChattyInfty to read, write and edit scientific documents using the speech output.[30][5]

Refering to the three coming technologies, they were introduced in the paper "Access to scientific content by visually impaired people" [5].

MathPlayer is a plug-in for Ms Internet Explorer3 introduced by Neil Soiffer in 2005. MathPlayer works with screen readers using Ms Active Accessibility (MSAA) interface. Screen readers are software applications to interpret what is being displayed on a screen. It actually sends to the screen reader a string containing the relevant sentence to read with a speech synthesizer. A MathZoom feature allows to magnify expressions. Built-in speech rules try to minimize the number of words used to speak the math.

MathTalk was developed by Robert Stevens, the MathTalk is another example of a system designed to render algebra to blind students. This system is aimed to students who are still at school, and is particularly aimed to those between the ages of 16 and 18. The design of this application is based on rendering the algebraic formula in a non-interpretive fashion. The MathTalk program uses a prosodic model to convey both the content and structure of algebraic formula.

The TechRead system was developed by Dr.Donal Fitzpatrick and Dr.Arthur I. Karshmer. The TechRead approach is aimed to produce a rendering of mathematics without using non-speech audio. Meaning it relies on the prosodic features found in synthetic speech to convey the material. The core idea involved the use of alterations in the pitch, speaking rate and amplitude of the voice to convey the content, and the introduction of placed pauses to convey the structure of the mathematics content.

2.5 Mathematica as a Choice

As Mathematica is our main program we are using and we are interested in exploring its notebooks. Knowing that it is a high-level programming language, this was the main reason behind implementing our application and its programming phase using Mathematica and make great use of all its privileges and capabilities. Our first approach was to do the implementation in any other language in C, C++ or Java and connect it to the Mathematica kernel through the MathLink. But writing the main code and implementation in a Mathematica notebook will require no link to the kernel. In-fact by just loading the package that includes the code will run the whole application at once.

Mathematica notebooks has also great capabilities, it is used as a great interactive notebook. As it interprets the input either its a text, mathematical formula or even graphics. It is capable of very good styling and formating documents. In case of programming, Mathematica is a high-level programming language, moreover the pattern matching style it has, known as "Rule-based programming style". A programmer can use this style to avoid complicated If-Then-Else statements. As the pattern matching works by checking on each name and type in one line of code, according to the defined patterns by the user.

As for "Blind People", Mathematica notebooks can be of great help. The speech commands that Mathematica allows, are of great use especially to blind people as they act as the speech synthesizers that blind people use to interact with computers and technical equipments. It simply works by typing command `Speak` and specifying what the user wants to hear enclosed in square brackets. One can introduce pauses and slots of silence in between the words to reflect to the user a more natural way of speech and to enhance the output to the user. Therefore in our implementation we advised users to close the speech synthesizers applications they are using to avoid interruption. As this functionality in Mathematica enriched us to avoid any external speech aid.

The style Mathematica notebooks is using to display mathematical content is the tree based technique. This way of illustration works better for blind people than the linear Braille display technique. Because it reserves the structure of the mathematical contents and preserves the space and effort of reading math in a linear fashion. This is reflected in our implementation, how blind users were able to traverse mathematical content in a tree form than a linear form.

Mathematica also allows blind users to interact with its notebooks using new equipments and tools. As it is discussed the GBM system in this content allows blind people to interact through touch gestures to perform certain actions. This could be applied to touch pads as I-pads, surfaces or even simple mouse pads. This technique makes us think out of the box, that Mathematica has capabilities and tools that can help blind users in many ways. Another possibility could be a speech input not only speech output, this approach would save a huge effort and burden on blind people.

The last thing which makes Mathematica a great tool for blind people is its capability to interact with Braille display. This is important to blind people because in this century most blind people consider Braille as the basic interaction paradigm for them exactly as sighted people consider the keyboard or the touch input as interaction method. Therefore when Mathematica allows the integration with Braille display this is a huge step for blind people. By simply connecting the Braille and dealing with Mathematica as any other program is a possible way of interaction, or even on a higher level one could think of connecting to Braille through the MathLink and this would get more control to the user.

At the end Mathematica revealed its capabilities the only thing is missing is someone to use these capabilities and make big use of all its powers. It showed how it could be of great help to visually impaired people, but one can extend his/her thoughts to cover help to all people with disabilities.
The next sections we will introduce the concept and implementation of the GBM system in more details.

Chapter 3

Specification and Development of GBM System

The main concept of the application is to facilitate an access to Mathematica notebooks using gestures and receiving speech as a feedback output. The state of the art of this thesis is not only in providing a new access to blind people, which is gestures and speech but also in the gesture detection algorithm itself. Allowing $1 algorithm to detect horizontal, vertical and diagonal line is a fresh new feature which is discussed in the coming sections in details.

3.1 Project Steps

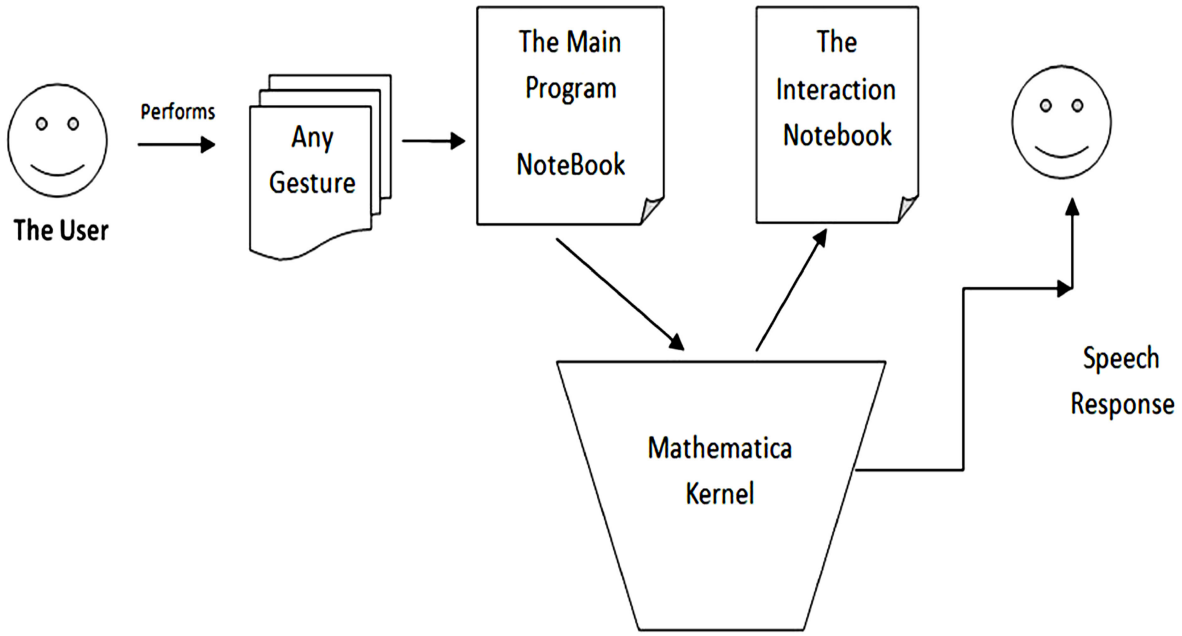


Figure 3.1: Project Steps

In Figure 3.1 we can see the steps of the project. The user initiates the interaction by performing any gesture from the defined ones. This gesture is read into the "Main Program" notebook which translates this gesture into an action, sends this action to the Mathematica kernel. The kernel performs this action on the "Interaction Notebook" and speaks the output of the action out loud to the user.

3.2 Project Structure

The project structure in its first approach as we can see it in Figure 3.2, was to implement the gesture detection algorithm in a C++ Program using Visual Studio this program is responsible for reading the gesture the user enters from the touch slate (any touch input device for example Iphone or Ipad) and through the "MathLink" which is the link between the Mathematica kernel and the C++ program the kernel will allow an action in the notebook the user is interacting with. This action could be to navigate in a cell, read its content or move to another cell ..etc. The link between the Kernel and any other notebook, external program or external device is what is called "MathLink", which allows the kernel to perform any task and run the execution of commands in it. The kernel acts as the brain to Mathematica, were all the tasks and commands are computed and executed in it.

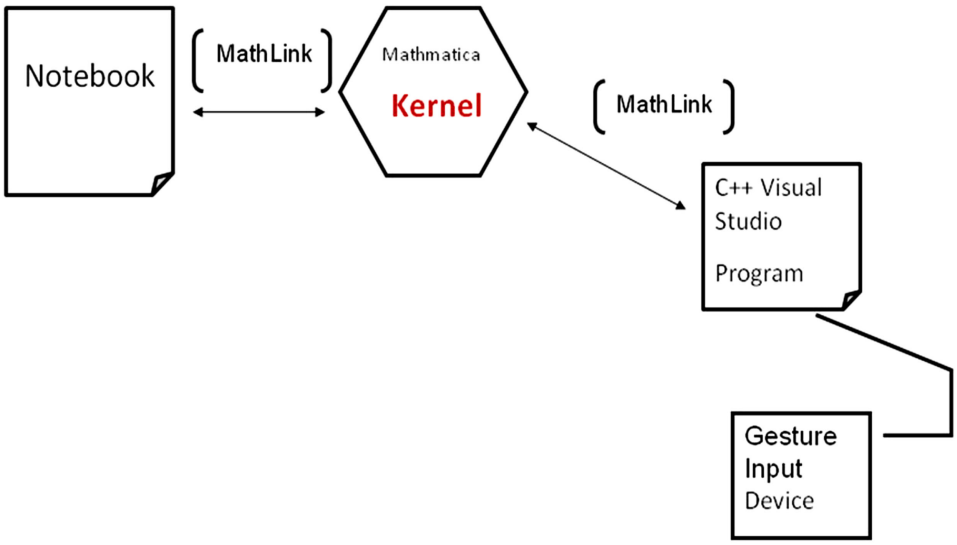


Figure 3.2: Old Projects Structure

The New project structure as we see it in the map below in Figure 3.3, reveals the true power of Mathematica. As it includes all the implementation of the application including the gesture detection algorithm in a unique notebook which is called the "The Main Program". The Main Program notebook is connected to the Mathematica kernel through the "MathLink", it reads the gesture input the user performs from the touch slate which is in the current state the touch pad embedded in any laptop. According to the gesture recognition the main program will send a certain action to the kernel. This action will be performed on the notebook the user is interacting with. As we can notice in the new approach everything is implemented and runs in Mathematica and depending on one brain in this case which is the Mathematica kernel. The advantages of this approach are obvious in its speed and the quick speech response to any gesture unlike the first approach which had many links and connections between different programs.

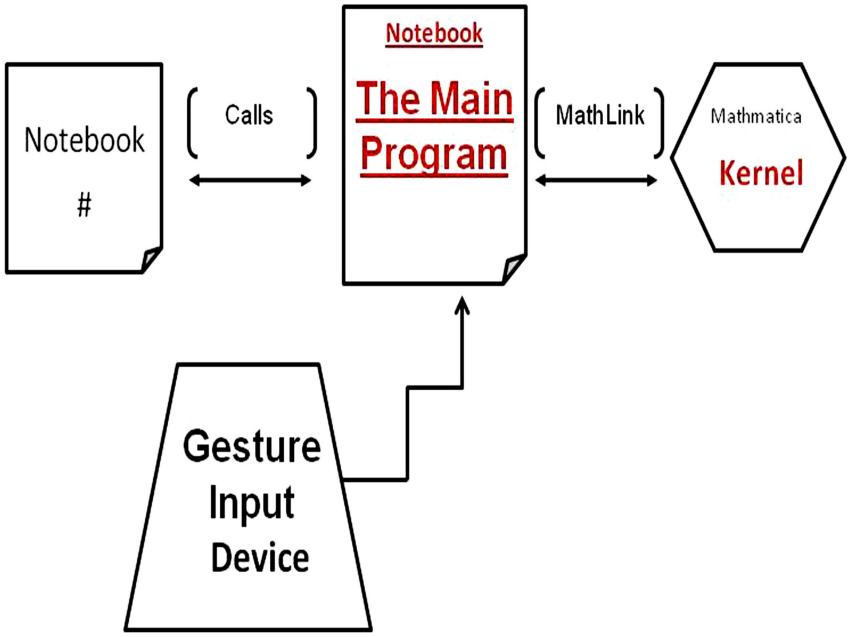


Figure 3.3: Current Projects Structure

Chapter 4

User Manual

4.1 Organization

4.1.1 Modes

This project has two modes:

- Practice Mode
  This mode is designed specially to help starting users to get to know the GBM system and practice the gestures and know their responses. Once they feel that they are accommodated to all the gestures only then they can move to the second mode.

- Interaction Mode
  This mode is the real mode were all the gestures perform actions on the notebook the user chose. At this mode the user starts the real navigation and an exciting trip into the notebook.

4.1.2 Button Notation

- "S"
  Started, enables the browsing field for the user to choose the interaction notebook.
- "E"
  Exit the whole process.
- "P"
  Practice Mode.
- "I"
  Interaction Mode.
- "M"
  Current Mouse Position.

4.1.3 Interaction Commands Gestures

As we had a look on the GBM system setup and the code overview in this section, now what is missing is how to interact with the GBM as a starting user.
The steps which were explained in the first part of this section how to open Mathematica and make the notebooks ready for interaction. Once these steps are completed the user would want to know how to interact with the GBM how to go through the notebook on a higher level and how to read the cells and make them speak on a lower level.

Knowing that Mathematica reads the functions and expressions in the form of nested lists were these lists represent nodes in a tree and the enclosed lists are their children in this tree. Therefore we are visualizing the functions and expressions as trees in our brains exactly as Mathematica does. Assume we have a virtual pointer which we use to browse this tree and traverse it. This is achieved by adjusting the position of this pointer at each node we want to speak out to the user. The template gestures that are defined in the GBM system, the user can use them to navigate through a Mathematica notebook, are shown in the Table below.

Gesture	Symbol	Abbreviation	Action
Left Square Bracket	[	LSqB	Go to the next neighboring left child on the same level of the tree.
Right Square Bracket	]	RSqB	Go to the next neighboring right child on the same level of the tree.
Diagonal Right Line	\	DRL	Go to the outermost right child of the current node.
Diagonal Left Line	/	DLL	Go to the outermost left child of the current node.
Vertical Line	|	VL	Go to a level upward in the tree.
Horizontal Line	—	HL	Explore the current node of the tree.
Caret Shape	∧	Caret	Switch between the higher and lower level of the Notebook.

4.2 Examples

The best way to explain to the user how to use the GBM system, or defining a user manual is through examples. As examples are the easiest way to explain the system, moreover mapping these examples to any other interaction the user wishes, will be quite easy.

4.2.1 The Higher Level "Notebook"

This is our notebook "Notebook X" with nested colored cells as shown in Figure 4.1. This example will show the user how to access "Notebook X" cells, cell groups and navigate through them.

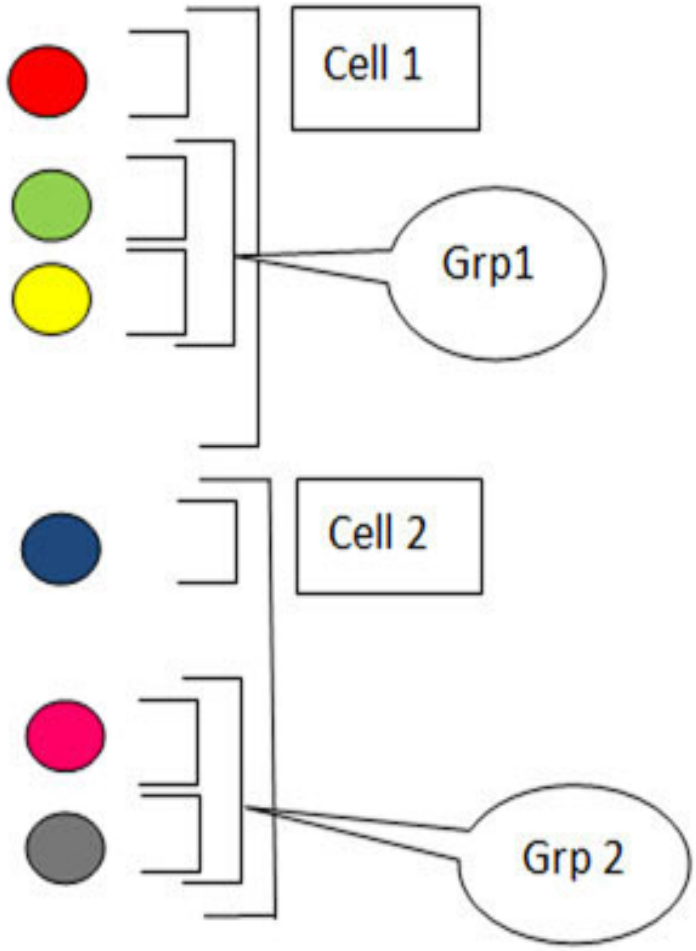


Figure 4.1: Notebook X, Mathematica Notebook Example

Figure 4.1 represents a notebook with two cells, we will refer to them as "Cell 1" and "Cell 2" respectively. The "Cell 1" consists of a single cell which is red and a group cell "Grp 1" with two other enclosed cells green and yellow respectively. The "Cell 2" in the notebook consists of a single cell which is blue and a group cell "Grp 2" with two other enclosed cells pink and gray respectively. This is the structure of the notebook with its cells.

The tree structure of the “Notebook X” as shown in Figure 4.2.

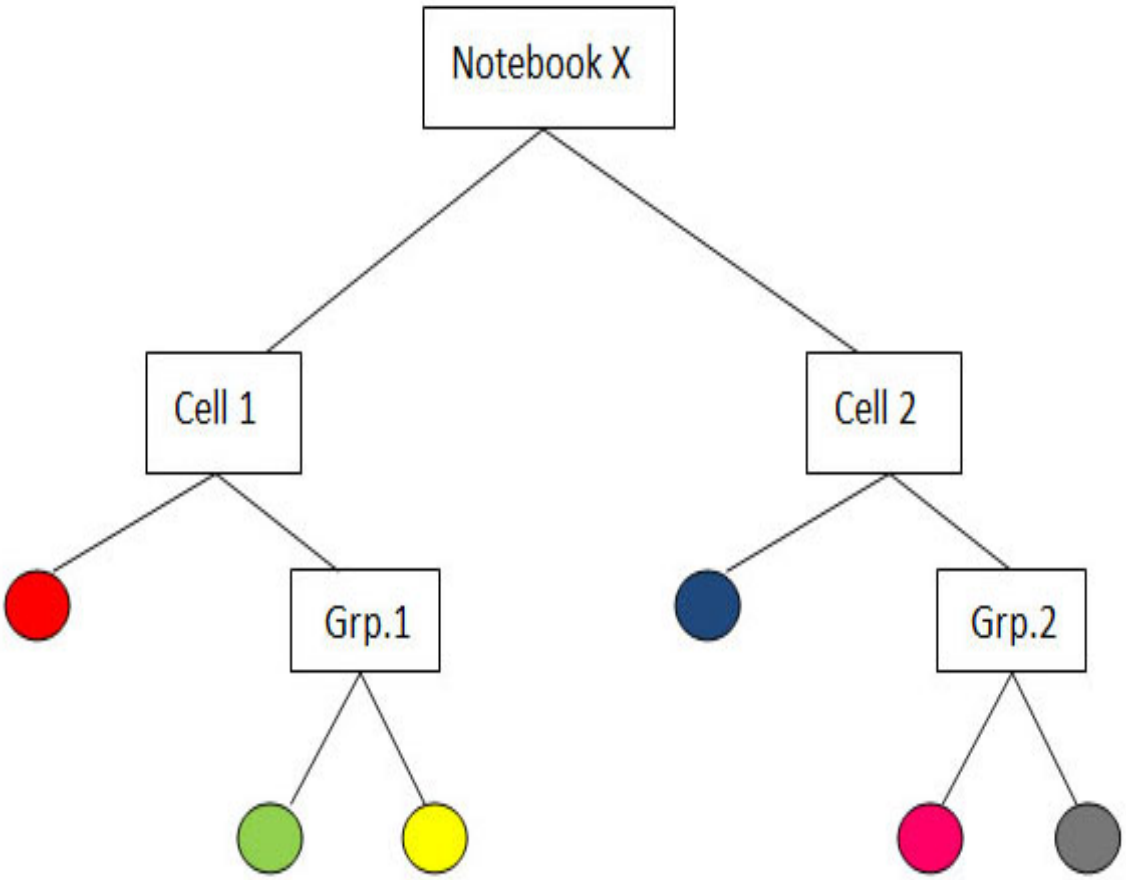


Figure 4.2: Tree Structure of Notebook X

Figure 4.2 represents a notebook in its tree structure, it has two children, we will refer to them as “Cell 1” and “Cell 2” respectively. The first child of the “Cell 1” includes a red leaf node, the second child of the “Cell 1” is a sub tree with a root “Grp 1”, this sub tree has two children the left child is a green leaf node and the right child is a red leaf node. Then we move to explore the content of the second child of the root node “Cell 2”. “Cell 2” has two children the left one is a blue leaf node, while the second one is a sub tree with a root “Grp 2”, this sub tree has two children the left one is a pink leaf node while the second one is a gray leaf node. At this point we finished traversing the notebook tree, next part will explain how to traverse it by gestures.

The table of navigational actions the user can possibly follow to visit all the tree nodes shown in the Table below.

Gesture	Action
HL	Speak a Notebook with 2 Children
DLL	Cell group data with 2 children
DLL	Speak type of red circle leaf
VL	Go Level Upwards
DRL	Cell Group data with 2 children
DLL	Speak type of green circle leaf
VL	Go Level Upwards
VL	Go Level Upwards
DRL	Cell Group data with 2 children
DRL	Cell Group data with 2 children
DLL	Speak type of pink circle leaf

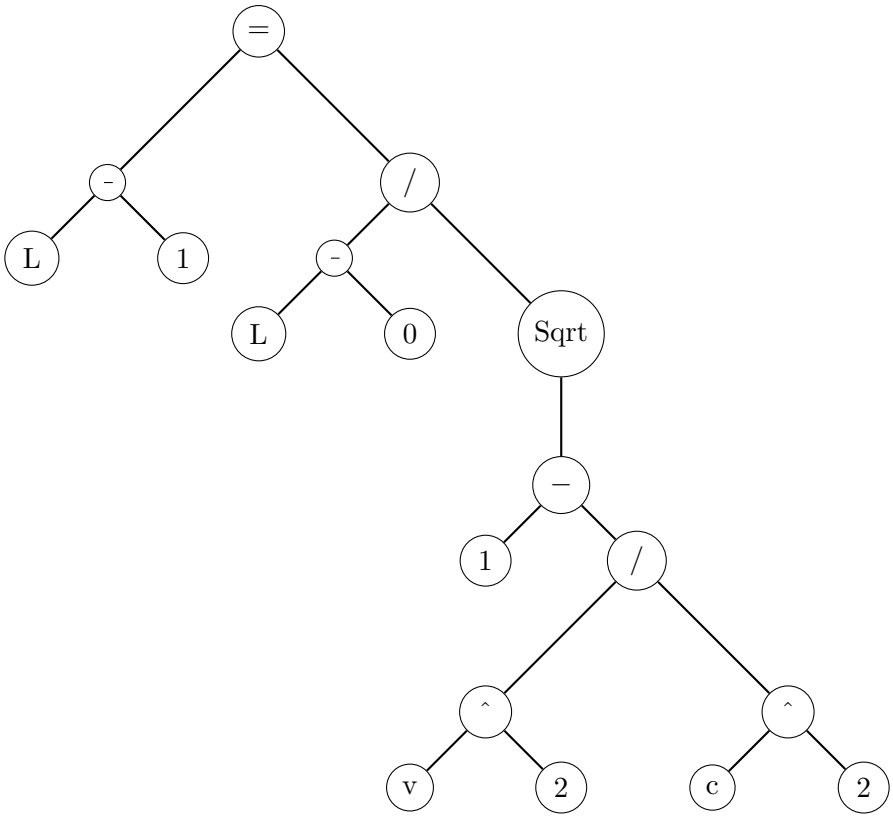
4.2.2 The Lower Level " Cell"

Referring to the formula we described before in the 2D form and in the linear form, as seen below respectively.

$$L_1 = \frac{L_0}{\sqrt{1-\frac{v^2}{c^2}}}$$

$$L_1 = L_0/Sqrt(1 - v\hat{}2/c\hat{}2)$$

We will see how can the user traverse the tree form of this formula in its linear form as seen below. We will see how a user can navigate through this tree and access its root up to its leafs step by step. The gestures that the user will apply and the action resulted in each case.



This is the navigational table in the cell level, each gesture results in a certain action which gives a specific output as shown in the Table below.

Step#	Gesture	Action	Output
1	HL	Stay at same Position	Speak 'Equal'
2	DLL	Go to left child node	Speak 'Subscript'
3	DLL	Go to left child node	Speak 'L'
4	RSqB	Go to right neighbor node	Speak 'One'
5	VL	Go level upward	Speak 'Subscript'
6	RSqB	Go to right neighbor node	Speak 'Fraction'
7	DLL	Go to left child node	Speak 'L'
8	RSqB	Go to right neighbor node	Speak 'Zero'
9	VL	Go level upward	Speak 'Subscript'
10	VL	Go level upward	Speak 'Fraction'
11	DRL	Go to right child node	Speak 'Square-Root'
12	DLL	Go to left child node	Speak 'Subtraction'
13	DLL	Go to left child node	Speak 'One'
14	VL	Go level upward	Speak 'Subtraction'
15	DRL	Go to right child node	Speak 'Fraction'
16	DLL	Go to left child node	Speak 'Power'
17	DLL	Go to left child node	Speak 'V'
18	RSqB	Go to right neighbor node	Speak 'Two'
19	VL	Go level upward	Speak 'Power'
20	RSqB	Go to right neighbor node	Speak 'Power'
21	DLL	Go to left child node	Speak 'C'
22	RSqB	Go to right neighbor node	Speak 'Two'

This section covered examples on both the higher and lower level of the tree. To provide the user with a practical image on how to use the GBM system.

Chapter 5

Implementation

5.1 Software Integration

To integrate the GBM system to get it working in other words, one should simply have a laptop with a built-in touch pad as in Figure 5.1, audio driver built-in or loud speakers and finally the most important thing Mathematica program installed on it.
The program was implemented on Mathematica version 8. It is also compatible with the new version Mathematica 9.

Figure 5.1: Laptop Touch-Pad

By having all these resources and the software code provided on a CD attached by the thesis, one can simply run the program. If the user wants to download Mathematica software for the first time one can purchase it from the "Wolfram Mathematica" home page in the link below:
http://www.wolfram.com/mathematica-home-edition/

The steps the user should make to get the program running for the first time:

1. Start Mathematica.
2. Evaluate the global variable `$UserAddOnsDirectory`.
3. This will tell you the name of the directory on your Mathematica installation, where Mathematica searches for user packages.
4. In this directory, there must be a subdirectory called "Applications".
5. Store the attached "GestureNavigation.m" in this "Applications" subdirectory.

To use the installed package:

1. Open Mathematica.
2. Open a new notebook.
3. Write Expression: Get["GestureNavigation`"] or Needs["GestureNavigation`"].
4. Evaluate the expression.
5. Press "S" to get started.
6. Choose the notebook you want to interact with by browsing through the files on the disk.
7. Press "I" for Interaction mode.
8. Press "P" for Practice mode.
9. Press "M" to Explore the mouse position.
10. Press "E" to Exist the whole process.

5.2 Code Overview

5.2.1 $1 Algorithm

The approach used in this paper is based on "$1 Algorithm for Gesture Recognition"; it was named after its privileges as it is easy to implement, doesn't require much coding and it is compatible with any UI one is using. We referred in our implementation to the pseudo-code provided at the end of the $1 algorithm paper. The pseudo-code is easy and straight to the point with every step explained, thanks to the publishers. [31]

Choosing $1 algorithm was based on many reasons some were already mentioned like its privilege being easy to implement, but also because it was compared to other algorithms as stated in the $1 algorithm paper which proved its efficiency. $1 algorithm was tested and with one loaded template supplied from the user it obtained 97% accuracy and 99% with three loaded templates. These results were compared to Dynamic Time Warping gesture recognizers which were proved equal and were compared with Rubine gesture recognizers which were much better. First we need to define Gestures; they are any movement done by the body to convey a certain intention as thoughts or emotions. The hands can act as an excellent tool for communication when one speaks; this tool is of great advantage in helping our users expressing the actions they want to reveal.

In this project we in particular deal with gestures as touch gestures. Through touching, they could express these actions using pen or fingers on touch slates like many new user interfaces for mobile, tablet, large display and tabletop computers.

Due to simplicity we will be using fingers and laptop touch pads. This makes it accessible to everyone who has a laptop and Mathematica installed on it, to use the GBM system.

This application is addressing blind, partially sighted and fully sighted people, as it is designed to help all people to listen to Math instead of reading it.

The simplicity of $1 algorithm is obvious in the shapes it includes; it includes very basic shapes in geometry as circles, triangles, square shapes and brackets for example.

One can notice the advantages of $1 algorithm, as it is invariant to rotation, scaling and translation. Invariance means that the gesture doesn't get affected by the transformations applied on it. Therefore $1 doesn't require any prior feature selection it is very flexible to transformations and can accept any change. Above all it results in a very high percentage recognition rate.

$1 algorithm restricts the focus of the user to uni-stroke movement which means the user moves one hand with gentle pressure over a surface typically once, these are defined as gestures that unfold over time.

Some of the gestures supported by $1 algorithm and used in this project are "Caret","V" and "Right/Left Square Bracket" shapes as seen in the Figure 5.2 below.

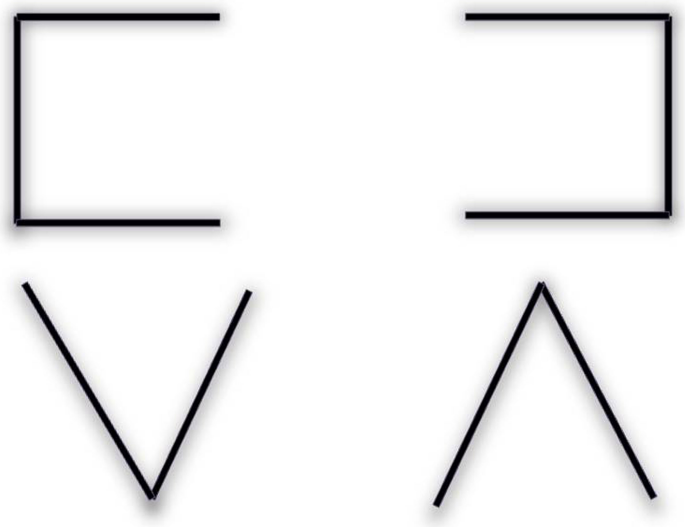


Figure 5.2: The shapes used in the project

5.2.2 The GBM Coding

The program starts by defining a set of four templates (Caret,V, Right/Left Square Bracket) shapes and stores them in a list $\{T_1, \ldots, T_n\}$.
By a left click on the mouse to initiate the gesture then drawing the gesture with one finger followed by a right click on the mouse the gesture will then be defined in the GBM system and entered in a correct way.
The gesture is stored as a list of points, each point has x and y position, we will refer to the list of points which represent the entered gesture as "C". This stored gesture "C" will be compared with the defined templates $\{T_1, \ldots, T_n\}$, the template T_i with the minimum distance difference from "C" will be chosen as the best match.

The candidate points are then sampled at a rate determined by the sensing hardware which is laptop touch-pads in our case. Putting into consideration the fact that human nature is different, some people draw gestures quickly and others draw them slowly which results in different number of points. People have different sizes of fingers which result in different sizes of gestures. Therefore the gesture entered by the user C will never find a perfect match in the defined templates $\{T_1, \ldots, T_n\}$. This is considered a challenge in the recognition process, to overcome this challenge one should think of a way to make the "C" find a perfect match from the list of templates. This is achieved by following the four steps $1 algorithm suggests, which is both the templates and the candidate points entered by the user have to be resampled, rotated once, scaled, and translated. This way the candidate points C will definitely find a perfect match which was treated in the same manner.

Before we start the steps of gesture recognition, to give an illustrative example one draws a gesture and enters it in the GBM system then tracks the changes that takes place in the

gesture after each step of recognition. The gesture drawn for testing is the right square bracket as shown in Figure 5.3.

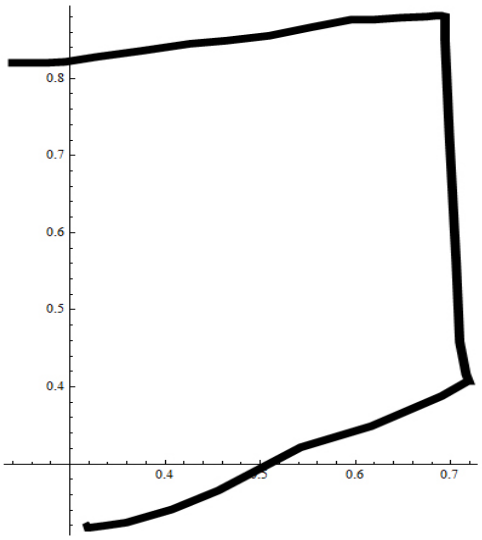


Figure 5.3: Right square bracket gesture

5.2.2.1 Step One: Resampling

As we mentioned before the movement speed is varying from a user to another, this challenge will affect the number of input points in the gesture. Therefore to make gestures comparable with each other even at different movement speeds, they should have the same number of points, therefore resampling will take place at this stage.

The gesture has M points which are the number of original points drawn by the user and the N points which are defined to be equidistant spaced points with the same distance from each other. N could be any number fixed for the templates and the candidate points ranging between $32 \leq N \leq 256$, we chose $N = 60$ in our approach. Beause number of points less than 32 points will not be that precice in forming the gesture shapes and points more than 256 will be so complex in computation and will affect the performance, these findings are based on the study in \$1 algorithm paper.

Applying the resampling by having M points divided by $(N - 1)$ = length of each increment, I, between N new equidistant points. Using linear interpolation a new point is added if the distance moved is greater than I.

Interpolation is a polynomial function of the line connecting a set of points (u), the degree of the polynomial function is usually $(u - 1)$. In our case we usually interpolate between two points on a straight line, therefore we have a polynomial function of degree one which we refer to as linear Interpolation. In this function we substitute by the two points to interpret

the point lying between them with a certain ratio. This is how we resample the shapes, re-arranging the 60 points on equidistant basis using linear Interpolation technique. Figure 5.5, illustrates the idea of resampling which is to preserve the same shape but with different number of equidistant points from each other. Therefore one can notice that the shape stays the same in the right square bracket example but the number of points changes each time we resample.

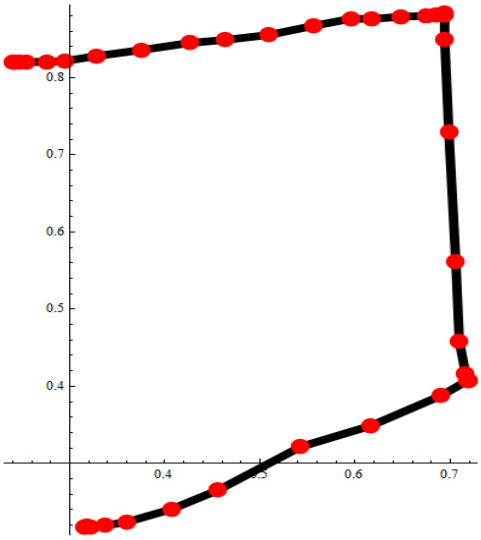


Figure 5.4: Square bracket shape without Resampling

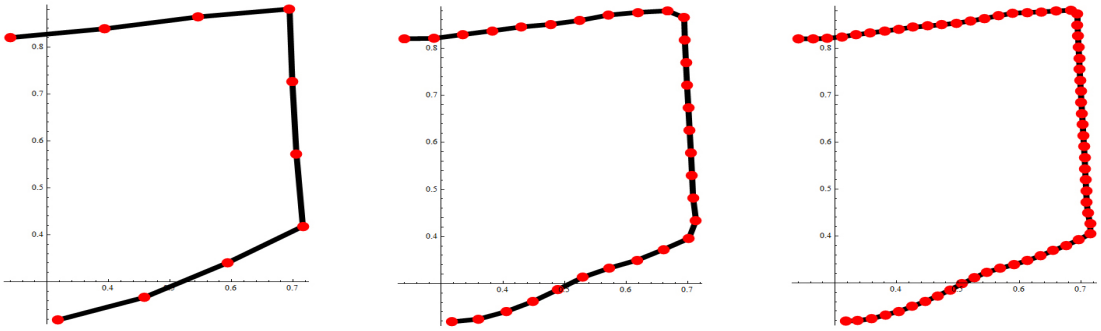


Figure 5.5: Resampling the right square bracket with $10, 30, 60$ points respectively

The resample function implemented in Mathematica, takes as an input the list of points and the number which we want to resample with and returns a list of points resampled with an equidistant space with a step I increments. Some rounding error may occur and the last point maybe skipped by chance due to difference in M points, therefore to overcome this problem we add it at the end, the algorithm is provided in the Appendix chapter part A "Resample Function".

5.2.2.2 Step Two: Rotation based on "Indicative Angle"

Reaching a point of having two paths of resampled points the candidate list of points "C" and the template chosen points T_i, there is no easy and straight forward way to choose an angle

which these two lists of points can be rotated with, to best align on each other. To overcome this issue the $1 algorithm introduces a way of searching over the space of all possible angles for the best alignment between these two set of points.

First, it finds the gesture's list of points "Indicative Angle", which is defined as the angle created between the centroid of the gesture (x, y) and the gesture's first point.
The centroid of any list of points is the average of points in this list. Finally, the gesture is rotated so that the centroid angle is at 0 ° in the Cartesian coordinate system.

As shown in Figure 5.6, the centroid of the right square bracket is estimated with the point enclosed inside it the red dot. Afterwords the shape will be rotated around the line between the first point of the right square bracket (any gesture) and its centroid, the line angle will be rotated so that it ends up at position 0 ° on the Cartesian plane.

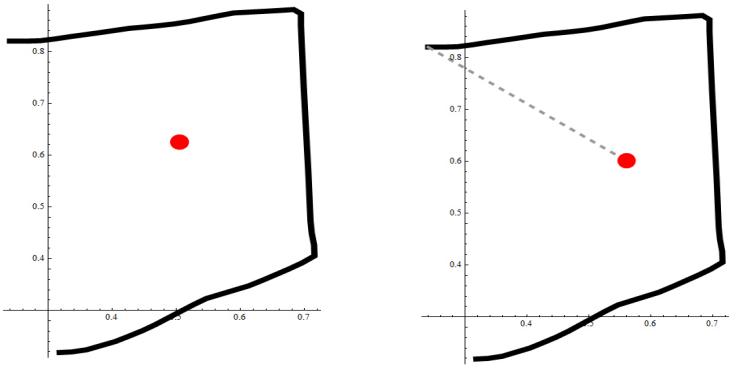


Figure 5.6: Centroid of the right square bracket

Figure 5.7, shows the effect of the "Rotation" function on the right square bracket gesture after the line connecting the first point of the gesture and its centroid is rotated, the algorithm is provided in the Appendix chapter part A "Rotation Function".

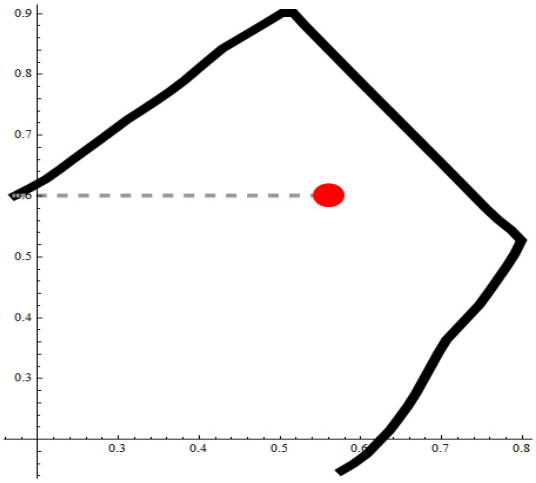


Figure 5.7: Rotation right square bracket gesture

5.2.2.3 Step Three: Scale and Translate

The gesture (List of points) is then scaled in a non-uniform fashion to a reference square. Bounding box, is the reference square which is calculated using the maximum and minimum points of the gesture list of points. As shown in Figure 5.8 the gesture is scaled with different scales according to the bounding box of each one, the algorithm is provided in the Appendix chapter part A "Scale Function".

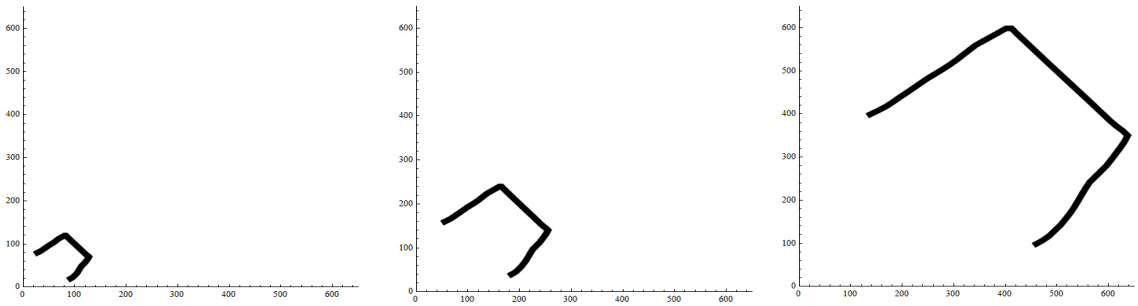


Figure 5.8: Scaling right square bracket with scale 100,200 and 500 respectively

Then the gesture is translated to a reference point, so that its centroid (x, y) is at position $(0, 0)$ as shown in Figure 5.9. The Figure shows the position of the gesture according to its centroid point before translation and after translation to the center. The algorithm is provided in the Appendix chapter part A "Translation Function".

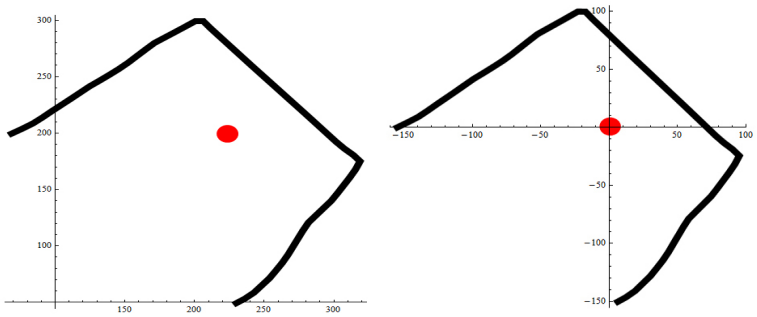


Figure 5.9: Right square bracket before translation and after translation

5.2.2.4 Step Four: Optimal Angle and Best Score

At this point the real recognition takes place were a candidate "C" is compared to each stored template T_i to find the average distance (Path distance) `di` between corresponding points "C" and T_i.

As we are comparing the two sets, the template T_i with the least path-distance to "C". The chosen template is considered the best match and of the recognition, this result is based upon the best angular alignment of "C" and T_i.

The question which should be asked, the algorithm already rotated the set of gestures once in Step 2 why do we need to rotate it again ?
The fact is that in Step 2, rotating "C" and T_i was once using their indicative angles which only approximated their best angular alignment but it still didn't reach the perfect alignment.

To find best alignment, "C" may need to be rotated again to find the least path-distance to T_i. The "Angular Space" which the angle lies in, must be searched for the minimum angle. This is achieved with the Golden Section Search (GSS), this algorithm finds the minimum value that the candidate "C" should be rotated by to align with a template T_i, using the Golden Ratio Φ=0.5(-1 + $\sqrt[2]{5}$). The golden ratio states that two values can be said to be in a golden ratio form if the ratio of their summation is to the larger value in them is greater than the ratio of the larger value to the smaller value as one can see in Figure 5.10. In our case GSS will be bounded by $\pm$ 45° and 2° as a threshold. The algorithm is provided in the Appendix chapter part A "Optimal Angle and Best Score Function". [32]

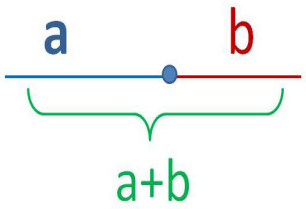


Figure 5.10: Golden Ratio: $a + b$ is to a as a is to b

5.2.3 Limitations of the $1 Algorithm

- It includes 2D space only no 3D shapes are recognizable.
- $1 algorithm cannot differentiate between shapes that differ in orientations, aspect ratios, or locations. For instance, to separate between squares and rectangles, circles and ovals, or directions as arrows directed upward or downward is not possible.
  Let us have an example to demonstrate the shapes different in orientations result in same recognition. If the user draws the right square bracket in the orientation A, as demonstrated in Figure 5.11.

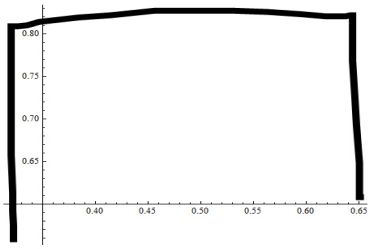


Figure 5.11: Right square bracket in orientation A

This gesture will go under the four steps of recognition which are resampling, rotation, scaling and translation. Afterwords this gesture will be compared to the defined templates in the system giving different scores the scores are simply the distance between the gesture and the templates, in this case the scores are: $90.97, 90.36, 80.88$. According to these findings, the system chooses the least score and the recognition at the end resulted in the right square bracket as shown in Figure 5.12.

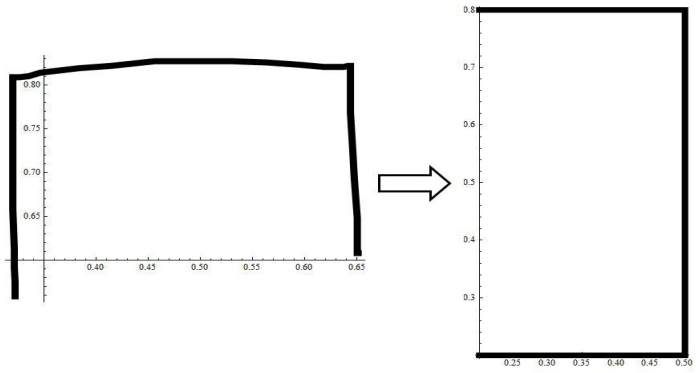


Figure 5.12: Right square bracket in orientation A resulted recognition

Now what if the user draws a right square bracket gesture in orientation B, as in Figure 5.13.

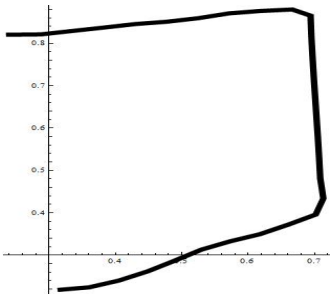


Figure 5.13: Right square bracket in orientation B

The gesture will go under same four steps as in the first case and the scores according to the distance difference between the gesture and the templates are: $95.73, 95.34, 84.10$. The system then again chooses the least score and the recognition at the end resulted in the right square bracket as in Figure 5.14. This example demonstrates that regardless of the orientation of the gesture the recognition steps that it goes through adjusts the gesture so that it results always in the same template.

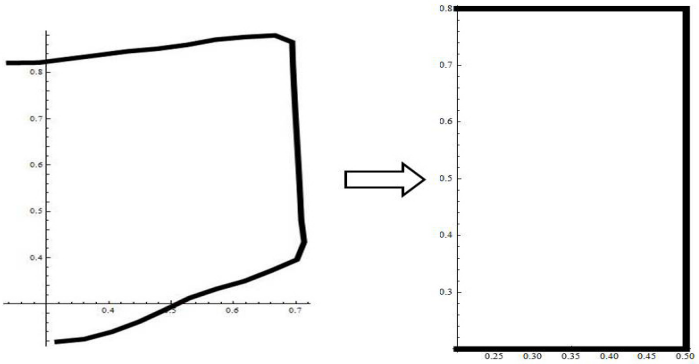


Figure 5.14: Right square bracket in orientation B resulted recognition

- Another limitation of the $1 algorithm is that horizontal and vertical lines are not recognizable. To demonstrate an example for this limitation, if the user enters a vertical line as in Figure 5.15, this line will go under the four recognition steps and the scores to the stored templates are: $101.97, 101.28, 85.09$. The system will choose the least score and will end in the recognition of the Caret shape as in Figure 5.15.

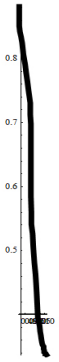


Figure 5.15: Vertical Line

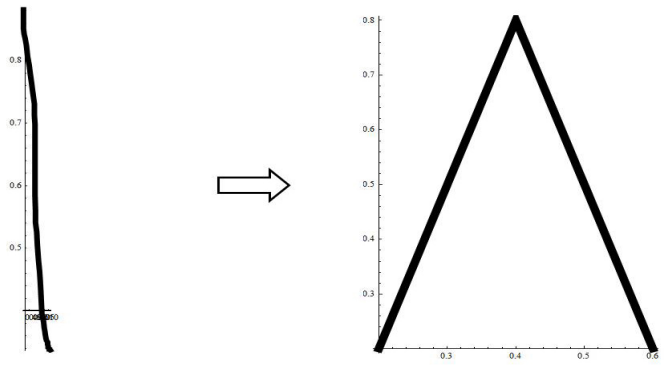


Figure 5.16: Vertical Line resulted in Caret recognition

Now if the user enters a horizontal line as in Figure 5.17, this line will go under the four recognition steps and the scores to the stored templates are: $88.38, 87.98, 83.37$. The system will choose the least score and will end in the recognition of the Caret shape as seen in Figure 5.18.

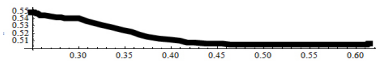


Figure 5.17: Horizontal Line

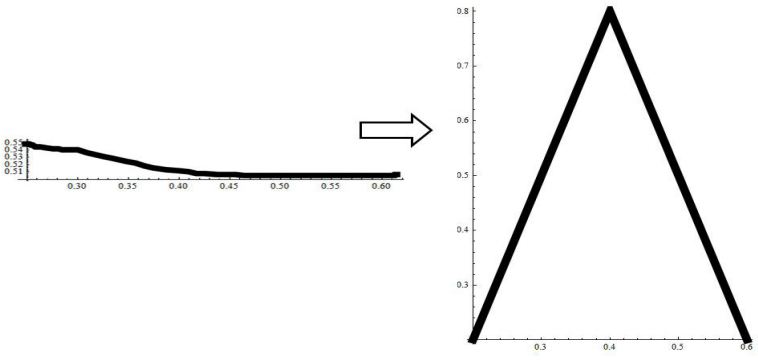


Figure 5.18: Vertical line resulted in Caret recognition

This example was able to show that whatever the gesture line entered by the user is, the result is always the same output as the system treats all gestures the same and won't be able to recognize the different orientation or direction.

- One would get surprised from the fact that the $1 algorithm also considers the starting point of the gesture as a crucial issue in the recognition process. If the user starts drawing the gesture from the right side rather than the left side, so that the starting point is on the right side of the gesture this will get conflicted with other gestures and won't be recognized in a correct way, it will simply be recognized as an opposite gesture being rotated. For example, if the user starts drawing the V shape from the right side it will be recognized as a Caret. As the system will perceive this as a Caret being rotated, not as a V shape being flipped or reflected. Also in the case of right and left square brackets. Therefore the start point in the gesture recognition process is a mandatory thing. As one can notice in the example in Figure 5.19 the V shape is started at the red dot after recognition the system estimated the perfect match template to be a Caret shape not a V shape anymore as in Figure 5.20. This reflects that still the user doesn't have the complete freedom to start the gesture from any point.

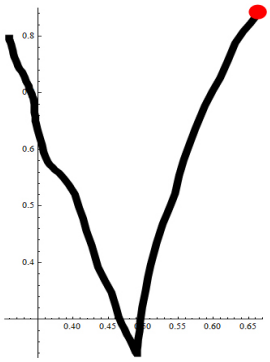


Figure 5.19: V Shape different start point

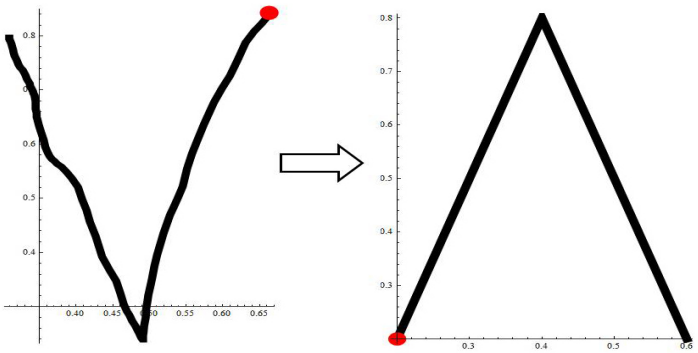


Figure 5.20: V shape perceived as a Caret shape

Another example is the right square bracket drawn from button to top is perceived as a left square bracket instead as in Figure 5.21.

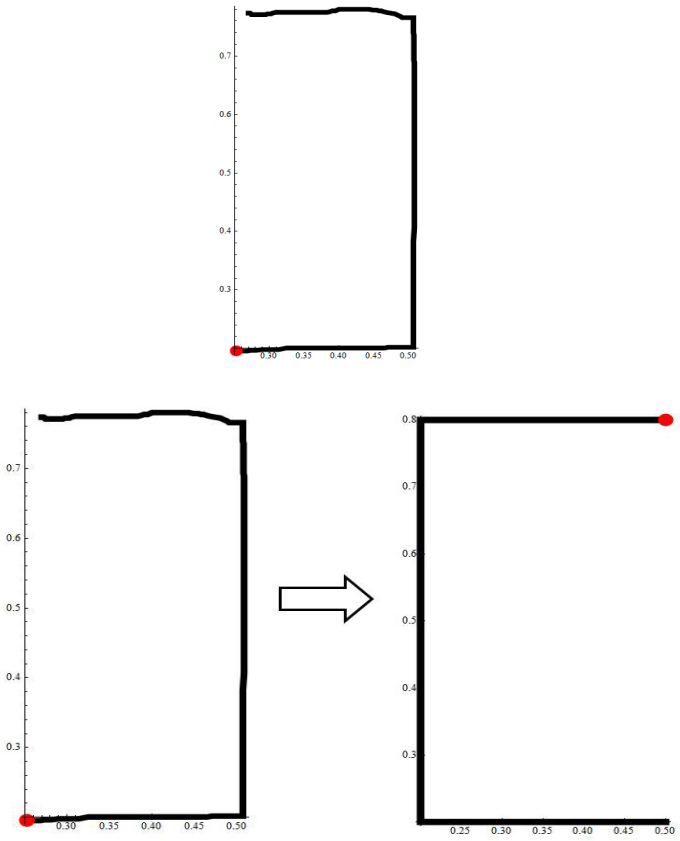


Figure 5.21: Right square bracket perceived as left square bracket

- At last, $1 does not include any time information, so gestures cannot be differentiated based on speed.

5.2.4 Enhancements on $1 Algorithm

As we mentioned in the limitations of $1 algorithm, horizontal and vertical lines are not recognizable. The reason is when these type of gestures are entered they have to be non-uniformly scaled using their surrounding bounding box this approach variances the recognition process because they are 1D not 2D gestures.

One approach to overcome the third limitation of lines recognition, is to test if the minimum dimension of the bounding box of the gesture exceeds a certain threshold, if this case doesn't occur this means the gesture is 1D and it is scaled in a uniform fashion. This results in a reference square around the gesture (e.g., line). Unfortunately, this approach is not supported in $1 algorithm.

From this point we define our new approach to introduce the $1 algorithm line recognition. Discussing with the user groups we discovered that the templates we specified [VShape, Caret,

Square Brackets] are not that easy and practical, they prefer uniform lines with different directions. Therefore our approach was trying to satisfy the users, we introduced in the $1 algorithm diagonal, vertical and horizontal lines as dynamic templates. But due to the many navigational commands we needed, we didn't omit the other templates we just mentioned. Figure 5.22 shows the lines supported in the new approach. [33]

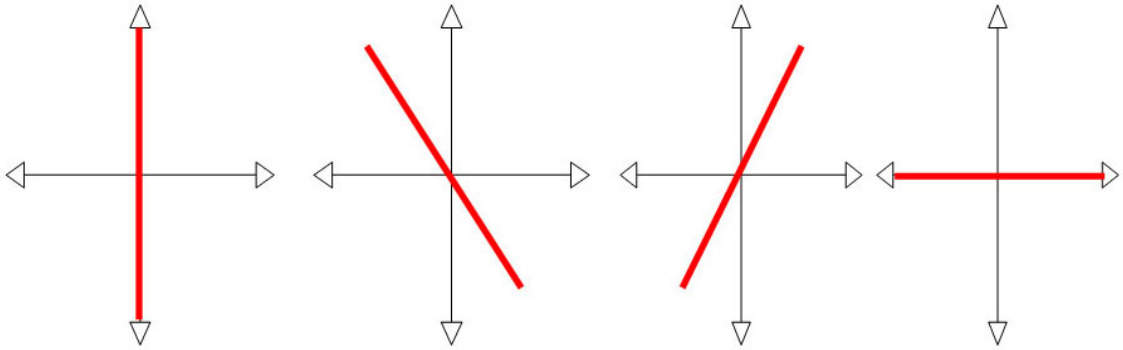


Figure 5.22: The type of lines defined, diagonal left, diagonal right, vertical and horizontal lines

If we had to consider the naive way we would have stored a line at each angle running from 0° to 360° degrees to cover all the possibilities of the line angles the user can draw.

We mean by dynamic templates, that they are templates that are computed by each entered gesture by the user, not pre-stored in the program as the fixed templates we had before {Square Brackets, V-Shape, Caret}. We do believe the dynamic way is more efficient to compute the best fitting line between a list of Candidate points "C". Therefore as the user enters a list of points gesture "C", the function which fits the best line between these points is called and it returns the equation of the best fitted line. We determine from this equation the starting and the ending points of the line by projecting the starting and ending points of the gesture "C" on it. Taking into account the starting and ending points of the line we construct a line between these two points, were it has an equidistant points as all the previous templates. As when we compare this line as a template it should contain the same number of points as the gesture "C" after resampling. It is resampled using the same number of points in our case 60 points.

This template is being stored in the list of templates $\{T_1, \ldots, T_n\}$ and the gesture "C" is compared to it and all the previously stored templates. The least distance between these templates and the gesture points "C" is being selected as the best match. If the best match was our template line we just constructed, then we get the angle of this line and decide where it occurs on the Cartesian plane and which line direction it indicates. The code of the line template construction is provided in the Appendix section A under "The Line Template Functions".

The only disadvantage to the dynamic template approach is its time consumption and speed limitations, when computing a dynamic template to each entered gesture.

The Figure 5.23, below illustrates the range of angles on which one can decide what type of line it is. If the user entered the red dashed line with angle α, therefore the system can

recognize the line according to the α location on the Cartesian plane. If α lies in the red region $110° \leq \alpha \leq 160°$ or $290° \leq \alpha \leq 340°$ then it is a diagonal left line. If the α lies in the blue region $20° \leq \alpha \leq 340°$ or $160° \leq \alpha \leq 200°$ then it is a horizontal line. If the α lies in the green region $70° \leq \alpha \leq 110°$ or $250° \leq \alpha \leq 290°$ then it is a vertical line. Finally, if the α lies in the yellow region $20° \leq \alpha \leq 70°$ or $200° \leq \alpha \leq 250°$ then it is a diagonal right line. The case of the red dashed line with angel α, it appears to lie in the last case the yellow region which is a diagonal right line.

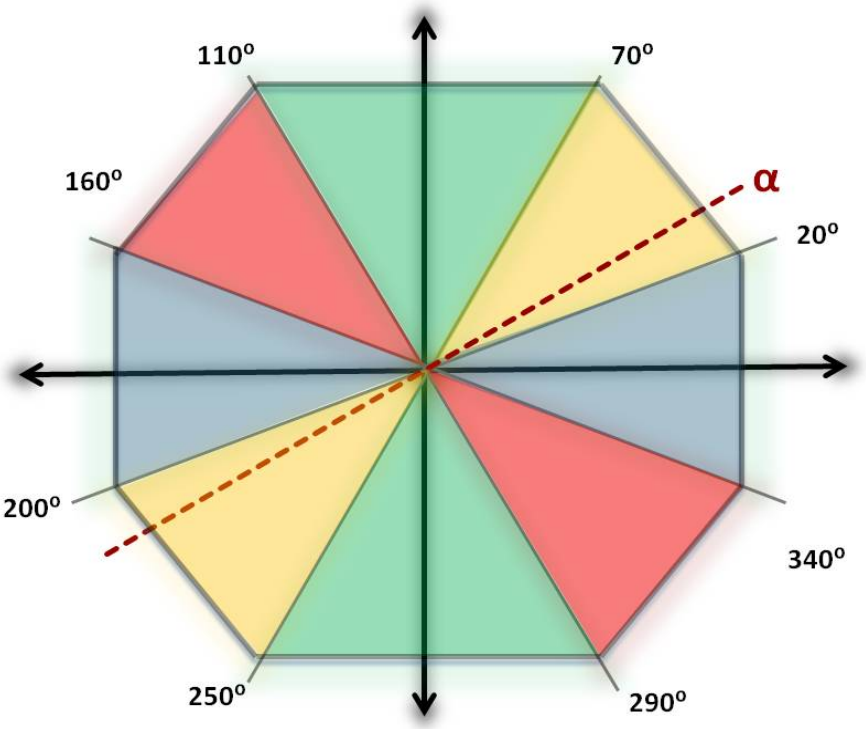


Figure 5.23: Range of Angles

At this point the gesture recognition is completed by returning a number indicating which template was chosen, from the template list including the dynamic template we just constructed, and finally according to the returned number a certain action is performed to navigate through the notebook.

5.3 The NoteBook Interactions

In the last part we managed to find the best template match to the input gesture by the user. The best match is represented as a number indicating the template chosen. This number is our main concern according to it the GBM system will do an action which will be spoken as an output to the user.

When people open any book, notebook, section or paper usually they like to browse it first, know the sections and the subsections it consists of, know the titles and subtitles and finally they can read it and go through it in details. This way of reading and exploration of the hierarchy before going in details is very beneficial, as it structures the piece of reading in the brain in the same way it is structured in reality. This makes the concept more clear to the user and shortens the distance between it and the human understanding.

Relying on this technique of reading and browsing interaction we will start our discussion. The user as a first step would have the opportunity to browse through the sections, subsections and titles and subtitles of the notebook before going into the cell contents.
The program is divided into two main levels:

- The Higher Level Interaction: Notebook
- The Lower Level Interaction: Cell

5.3.1 The Higher Level "Notebook Level"

The notebook the user chooses and wants to browse can be treated in the form of a tree structure, were the root of the tree is the notebook list with all the cells and the cells are the children of this root. The user can apply the set of gestures to access the tree nodes, for example drawing a diagonal left line to access the left child of the root node.

The function `StartNB[]` acts as the wheel of the project, as it initiates the starting commands and initiates the notebook the user wants to interact with. The algorithm of this function is provided in the Appendix chapter section A, "Start Function". The user opens a notebook to interact with and navigate through by pressing "S" and then chooses the notebook path from the browsing list appearing. This path will be stored in `Path`, and the notebook will be opened. As a first step we store the content of the notebook in `NBInteraction`. Then the selection is moved to the `GestureNB` as this is the notebook the user is drawing the gestures at, this step was mandatory because the users couldn't estimate the drawing boundaries. Therefore the best solution was to make no boundaries and let the screen contain the drawing area alone by hiding all the menu bars and the interaction notebook the user chose.This is handled by specifying the full screen option in the drawing area of the `GestureNB`.
The initiation of the Position in the current notebook is set to a list `PosNB = 1`, pointing to the root of this notebook. The flag `$atNB = True;` is set to true which activates the notebook Interaction function as a default case. Based on the assumption that the user would like to explore the notebook the higher level first then move to the lower level, the "Cell Level" this flag will be converted to `False` by then.

Taking into consideration the two levels we are interacting with. The tree structure of the notebook is quite different from that of any other tree, as the notebook tree relies on the pattern matching concept in Mathematica, were the nodes of the tree can lie in one of three categories:

1. List of cells:`{__Cell}`
2. Cell Group Data:`Cell[CellGroupData[_List, status_]]`
3. Cell with Input / Output:`Cell[_, _String, ___]`

Each time the user performs an action the current position in the notebook will lie in any of these cases. The switch command applies the pattern matching concept, it will compare the current position in the tree and will select one of these cases.
In each case the user can perform further actions according to the category the position lies in. For example, the user can explore what position the pointer is standing at, by performing a horizontal line, or move to the left or right child of this node, these actions will require moving to another position in the tree and it will result in a different case and match to a different pattern from the three categories we just mentioned.

The notebook Interaction function is activated as long as the flag `$atNB` is true, then the content of the notebook of the current position is being extracted and stored in `exp`.
A `Switch []` command will take place to evaluate the three categories and decide which one will take place according to the position in the tree `PosNB`.
Before going through the three categories let us have a deeper look on pattern matching in Mathematica.

5.3.1.1 Pattern Matching

Pattern matching in Mathematica represent different expressions and different categories and classes. Some expressions may result in the same answer mathematically, but their patterns don't match. Because their structures are not equal, for example: $1 + 2 == 3$ and $7 - 4 == 3$, both give the same answer 3, but if we check them upon pattern matching they will give a false result because $1 + 2 \neq 7 - 4$.

Therefore we conclude that Mathematica is based on structure aspects in pattern matching not on algebraic aspects, this is the core issue in pattern matching in Mathematica.

Let us see some examples of pattern matching:

Double blanks: Sequence of one or more expressions.
Triple blanks: Sequence of zero or more expressions.
`X_List  Or X{ _ _ _ }`: A list
`X:{ _ _ _ List} Or X::{{ _ _ _ }.}`:A list of many lists
`X:{{ _,_ }}`:A list of pairs
`X_^ n_.`:X power n, were n not equal zero

Other form of pattern matching is what so called "Classes of Expressions". These types of classes are made to generalize the pattern matching. They sometimes specify names to some expressions which allow the implementer to refer to them.[4]

`f[_]`: Any expression of the form f[anything].
`f[x_]`: Any expression of the form f[anything], with a name reference to this expression.
`F[A_,B_ ]`: F with named two arguments A and B.
`F[X_,X_]`: F with the same two arguments.

Pattern matching is a way to ease the matching and picturing of expressions based on their structure not their meaning algebraically or geometrically. This is a powerful option in Mathematica and eases the matching process on the implementers.

5.3.1.2 Four Pattern Matching Categories

An overview of the four categories the notebook interaction tree can lie in, the code is provided in the Appendix chapter under "Pattern Matching Cases".

1. This category lies in the case of having a list with any number of cells. This case usually takes place as the user starts the notebook at the beginning, when the `PosNB` points to the root of the notebook. `PosNB` is a list which acts as a pointer in the notebook level, as it holds the current position in the notebook. The expression `exp` is the node itself and its content, its length is computed to know the number of cells this notebook holds. A prompt is spoken out to the user saying "A Notebook of " + number of cells it holds.
   The user in this case will have the opportunity to move to the outermost left child or to the outermost right child, by performing specific gestures for these actions as the cases in left diagonal line and right diagonal line respectively. In these cases the `PosNB` list is being updated by appending a "1" in the outermost left case or appending the length of children of the notebook "l" in the outermost right case. At this point the `PosNB` will be pointing at a new child node in the tree.

2. This category lies in the case of having a Cell containing a `CellGroupData` of any number of cells inside this group. This case is the closest case of having a normal tree with a number of children, were the root of the tree is the `CellGroupData` and the children are the cells enclosed in this group.
   In this case the user will have the opportunity to move to the outermost left/right child or move to the left/right neighbor node on the same level or move to an upper level in the tree.
   In the first case moving to the outermost left/right child the `PosNB` has to be updated by joining `(1,1,1)` and `(1,1,'Length of children at this node')` respectively, so as to move a level downward in the tree.
   In the second case the user wishes to access the neighboring nodes, the `PosNB` will be updated by adding "+1" or "-1" to the last element in the list to get the right neighbor node or the left neighbor node respectively on the same level of the tree.
   The last case if the user wishes to go one step upward in the tree and move to a higher level, the `PosNB` will be updated by dropping the last three elements in the position list.

3. This category lies in the case of having a single cell with a String type. This string could be an Input/Output or a title, section or subsection form. In this case the user has the opportunity of accessing the right/left neighboring nodes in this cell by updating the `PosNB` by adding "+1" or "-1" respectively. The user can go one level upward by dropping the last 3 elements in the `PosNB` list.

Finally, if the type of the cell is an Input/Output then the user has the option to switch to a cell interaction technique at this point. If the user switches, the content of the cell is being extracted in the form of an expression and stored in the global variable `CellExp` and the flag `$atNB` is set to false to activate the cell interaction function.
In-case the type of the cell is a title, section or subsection then it is spoken to the user without switching to a cell interaction, assuming that the user wouldn't interact with a static text except listening to its content.
This introduces us to a very important constraint in the type of notebooks this program supports; it supports only simple notebooks and doesn't allow mixing text with mathematical contents, because these cases are not handled.
For example it doesn't support : Today I had a very difficult math problem in class where the teacher wanted us to figure out the value of $1/x$ based on some equations where x ...etc.
As in Mathematica these types of texts has another category in pattern matching which wasn't supported in this program.

4. The base case is the default case that is being checked each time, this case holds the possibility that the `PosNB` includes "-1" which is out of the notebook boundaries. If this case takes place the user is being notified that there are no children at this current position of the notebook and the position of the `PosNB` is being preserved before this error takes place. Another case handling if the user wishes to access a child's node were this node happens to be a leaf, then the user is also notified that he/she is accessing a node with no children and the position in preserved before this error occurs as well. This step is an error handling and giving notification to the user so as not to step out of bounds.

5.3.2 Lower Level "Cell Level"

As we want to move from the higher level and go further to lower level to the cell level, the user can do this switch gesture command which adjusts a flag `$atNB` to allow the interaction over the required level and activates the required interaction function.

As we mentioned before this switch command sets the flag and also assigns to the global variable `CellExp` the current expression the `PosNB` was pointing at.
The user can browse and interact with the values in the `CellExp` as long as the pointer position is in the current cell. The user can explore the cell and its content as a tree, access the tree children and move to its leaves. And as the user moves to another cell the `CellExp` is updated and the position in the cell is initiated as soon as a new cell starts.
The switch case is checked with the flag value `$atNB`, by each input gesture the user enters. Each gesture is encoded with an integer number, which is recognized inside the implementation code and according to this integer an action is being performed. The user has many possibilities of interaction in a single cell as shown in the cases below.
As the `CellExp` is assigned a new cell expression, a pointer is initiated to point at the position in the cell and treats it as a tree which traverses its nodes, the pointer inside the cell is also a list and referred to as `PosCell`.

5.3.2.1 The Cases

If the integer passed from the gesture recognition functions is encoded to be:

- "Horizontal Line"
  This will return as an action the node the user is currently pointing at.

- "Diagonal Left Line"
  This will return the outermost left child of the current node in the specified cell. It appends to `PosCell` a "1" to the current position in the specified cell to update the pointer and move a level downward.

- "Right Square Bracket"
  This will return the neighboring right node of the current node in the cell, to allow the user to access the neighboring right nodes, by adding "+1" to the last element in the pointer list `PosCell`. This will increment the position of the pointer to jump once from a child to another on the same level of the tree.

- "Left square bracket"
  This will return the neighboring left node of the current node, to allow the user to access the neighboring left nodes, by adding "-1" to the last element in the pointer list `PosCell`. This will decrement the position of the pointer to jump once from a child to another on the same level of the tree.

- "Vertical Line"
  This will allow the user to go a level upward in the tree in the specified cell, by dropping from the pointer list `PosCell` the last element.

- "Diagonal Right Line"
  This will return the outermost right child of the current node in the specified cell. By only appending to the pointer list `PosCell` the length of the children of the current node.

- "Caret Shape"
  This allows the user to switch between the cell level and the notebook level and vice versa. And it sets the flag `$atNB` to `True` to activate the higher level Interaction function. There are two interaction functions defined in Mathematica code one for the notebook level and the other for the cell level, when a gesture is drawn and submitted to the system, it automatically checks the flag `$atNB` and according to its value, the system decides which interaction function to enter and on which level to navigate.

- Base Case
  The base case is to extract the content of the cell at the current pointer position list and check if it doesn't exists then the user is notified that he/she is trying to access an empty node. Also if the pointer position list `PosCell` reaches "-1" the user is notified that the cell boundaries have been crossed. If the pointer position list `PosCell` is an empty list this indicates that the user is at the root of the cell and the whole content of the cell is spoken out.

These are the cases the user can perform in both the higher and the lower level of the notebook. Each gesture has a specific meaning and a specific number, the number is sent back to the GBM system when this gesture is performed. Regardless of which level the user is standing at and at which position the pointer is pointing to, these gestures are defined for certain actions. At this point the implementation phase is over, in the next chapter we will have a look on the evaluation and user feedback from the GBM system.

Chapter 6

Evaluation

6.1 The Evaluation Methodology

The testing methodology that was being used to test the GBM system was simply to allow sighted and blind users to try using it and making great usage of all its functionalities. This step was so important because it introduced some errors that were handled after the testing phase, moreover some insights and functionalities that were added as discussed in the next section. The user were introduced to the system and its functionalities as a simple explanation how they could use it to traverse Mathematica notebooks. Afterwords they were introduced to the practice mode, were they trained themselves on the gestures and they knew the action associated with each gesture. As an advanced step, they moved to the interaction mode were they had a testing notebook to traverse it using the system and explore the speech output. This was the testing performed, in the current phase.

Figure 6.1: User testing the laptop touch-pad

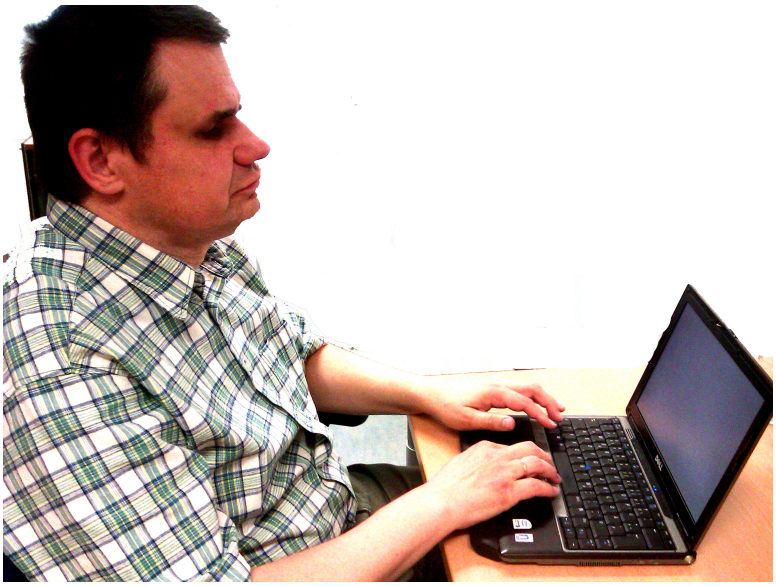


Figure 6.2: User testing the gestures and receiving sound feedback

6.2 Users Feedback

This section is dedicated to show the results of the GBM system during the test phase and the user's feedback. The overall GBM system feedback was pleasant, the state of the art was applicable and joining touch gestures to sound feedback turned out to be successful. The gestures that were used were easy to learn and to memorize by the users, adding the lines to the gestures was of a great help as they are easy to draw and memorize. The speech output was good enough for the current phase. The user was able to navigate through test notebooks and understand their structure, which is our main goal.

But any testing phase could have a bright side and a dark one, therefore as the users started using the GBM system, the sighted people didn't face any problems during testing, but some issues appeared to be uncovered while testing with the visually impaired people.

6.2.1 Challenges

The visually impaired people requested a practice mode to train themselves on the gestures and to memorize the output and action of each. Therefore we made this affordable by pressing on "p" as the GBM system starts. This will make the whole project run in the practice mode, the user will draw the gesture which will result in its name as a sound feedback, but without doing any navigational action in any notebook.

One of the major issues that appeared, was that while the users were drawing the gestures and clicking on the mouse, they accessed the menu bar of Mathematica by mistake, even if the gesture drawing notebook was made to full size this couldn't overcome the problem. Therefore a solution to this was to let the gesture drawing notebook in a "FullScreen" mode and nothing appears above this notebook, so that the selection will always be to this single notebook. This

will prevent the mouse from clicking or accessing anything not desired from the user's side. The user can simply exit the full screen by pressing on "E" button on keyboard.

Another issue was noticed during testing is that the users find difficulty in sticking to the boundaries of the touch pad. Which results sometimes in bad recognition, the solution for this is to replace the small built in touch pad of the laptops and use a large portable touch slates instead, to give the user the free space for interaction.

The initial mouse position is quite a challenging issue, as after the user draws the gesture he/she becomes lost where the mouse position ended up. When they try to draw another gesture, they most probably end up at the screen boundaries and the operation is not performed successfully. We are here addressing a serious problem, as the user now main focus is directed to the mouse position not to the main aim of the GBM system which is notebook navigation. The idea became more mouse oriented than gesture oriented, which is not our target. A solution to overcome this problem was to provide the user with a guide or a lamp in a dark cave, the button "M" provides the user with the current position of the mouse and guides him/her where to move the mouse in order to initiate a new gesture. The Figure 6.3 explains the organization of the mouse motion, if the mouse position lies in the "South East" then the user is advised to move a little bit upward and to the left, if the mouse position lies in the "South West" then the user is advised to move a little bit upward, if the mouse position lies in the "North East" then the user is advised to move to the left and finally if the mouse position lies in the "North West" then the user can draw the gesture as the mouse position is standing at the initial stage.

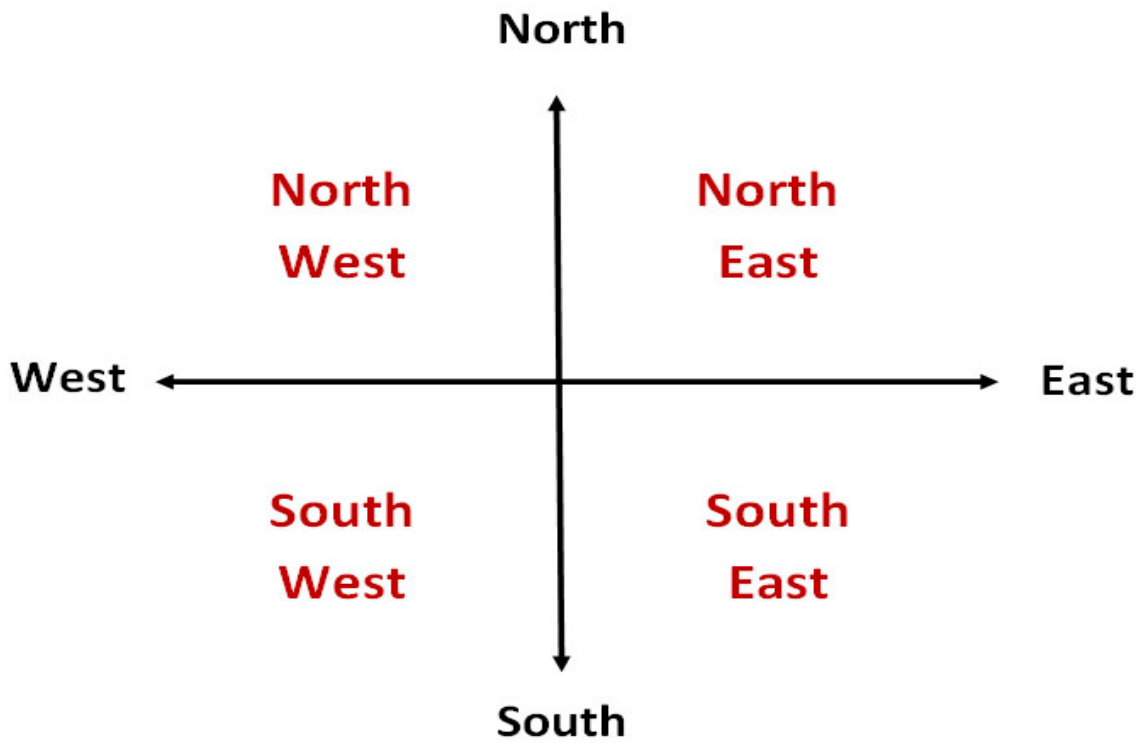


Figure 6.3: Mouse Position Possibilities

Testing the GBM system on different notebooks, one was able to discover that any text should be plain text no styling is supported by the GBM. Also no mixture of text and mathematical content is supported; this could be introduced in the future.

During testing and setting up the GBM system, it took some time to enable the Braille display during testing. Therefore it is recommended to close the Braille display and the screen readers as "Cobra Screen Reader" while using this application. A link is still not established between the Braille and the GBM system. Although the speech engine of the Braille display may still be connected and replaces the Mathematica speech which of course is preferable from the user's side, as they are used to it. But still this makes the performance slower, therefore we recommend disconnecting it.

Chapter 7

Future Work

Addressing future work in the field of blind people and mathematics will put a lot of flash lights on many ideas and enhancements that one can add. This field is considered new, it doesn't run to centuries in the past. Most of the applications that took place can be further adjusted and enhanced. We will have a look on the future direction in this field and then speak specifically on the enhancements that need to take place to our GBM system.

7.1 Insights Into the Future

Admitting that the mathematical Braille notation is so complex on blind people, due to the fact that it is linear this opens many ways to people to think how to simplify this notation. The MathML language is a language intended to describe the mathematical notations its structure and content. It is simply an application for XML, therefore it is used to integrate the mathematical formulas and notations into the Web. One of the future research fields is to render this language on Mozilla. This project will help blind people access the mathematical notations on the Web easily.[10]

One of MathML applications is the Mathematical Working Environment (MAWEN) it is in the development phase. In cooperation between the Johannes Kepler University of Linz, Austria, and the University Pierre et Marie Curie in Paris. The main aim of this application is to support documents that include text and mathematical notations, this issue we lack in our project. It supports navigation through formula by collapse and expand option, this option adds more view and deepens the understanding of the blind users, it is applicable for both views for sighted and blind people. It also allows editing the mathematical content in the two views. The system includes other features which are still under development, but it was made to address most of the problems blind people face.[5]

MathType is an equation editor which allows the users to write and formulate mathematical notations for Word, LaTeX, TeX, MathML and web pages. These editors make mathematical access much easier for sighted and blind people as well, it saves the effort of the Braille mathematical notation. One of the examples of these editors is the Math Genie.[5]

One of the issues that researchers should focus on, is the link between sighted and visually impaired people. Interfaces that serve both needs and establishes a link between the two parties is so important. Imagine a sighted student and a blind student who want to work together on a project or blind students who want to interact with his/her sighted teacher, this must be possible. Moreover the mathematical computations can be organized by a sighted person in a way which is easily navigated by other sighted people, but are quite hard for blind people. Therefore we need this link to be established, and existence for such editors that supports Braille notation and applicable for sighted people as well. We are eager to decrease the gap between a blind student and his/her sighted classmates.

AudioMath is a fresh concept in this field as well, which is the speaking math. As we have agreed that speaking math is different than speaking normal text, because speaking math should reveal the true structure of the content and apply the pauses in the correct places. This makes math much easier for those who listen to it. This field is increasingly growing, and on the future research directions it needs to be integrated with the MathML to support reading math with HTML,XHTML and other languages on the Web. It should provide navigation in mathematical content on web with a spoken feedback.[34]

This will bring us to the field of the prosody of reading mathematics. The prosody refers to the properties of speech, properties like the core meaning of the text which reflects what the reader is feeling, the emotions or the intensions by the speech. One could have a question or a sarcasm phrase which if spoken by a monotone style, its true meaning will never be reflected to the user. These properties could be revealed in the voice pitch, how the segment of the word could be elongated or shrinked, stress in words and the loudness of the voice. The difficulty in predicting the prosody of a still text is always a challange, because it doesn't depend on any rules or grammar, it is something understood and interpreted from the content. The same difficulty appears in the mathematical content; it increases as the interpretation of the speech in some cases leads to wrong answer or wrong understanding to the user. Therefore there is a huge amount of research going on in this field.[35]

7.2 Future Work Enhancements to the GBM System

7.2.1 Functionalities

This part states the functionalities that the GBM system couldn't cover in the mean while. Therefore we introduce them as future work to be added and integrated to the GBM system.

- Collapse and Expand
  The collapse and expand functionalities are important in the GBM system. Because as we know that the GBM system is dealing with Mathematica notebooks, therefore adding this functionality can add a feature to the user to group cells and ungroup cells. For example as the user is done navigating in part of the notebook, this part can simply be collapsed and a new part can be further expanded. A suitable gesture could be to use two fingers and move them toward each other in case of collapse or move them away from each other in case of expand. This gesture is applied in the recent touch technology systems, when a user wants to zoom into the screen or zoom out of the screen the same concept of collapse/expand takes place.

- Combined Text
  We mean here the text that includes normal text and mathematical functions, we refer to as combined text. Unfortunately the GBM system in its current state doesn't support such type of text. Due to the fact that their cases are not handled in the pattern matching part.

- Styled Text
  As adding any style to the text would differ in the structure we specified and it won't match with the pattern matching we defined. Therefore the notebooks should include plain text with no styling. This is to be enhanced further to differentiate between the text and its style and to be able to speak the text part only to the user.

- Mouse Click
  During the user tests, an issue has appeared which somehow lessens the speed and functionality of the GBM system. As the touch input is a perfect idea for visually impaired people, the way the gesture is being input is kind of tiring to the user. To press on left mouse button to start the gesture follows a right mouse click to enclose the gesture is somehow confusing to the users and distracts their focus. Therefore as a future research, one could enhance the start and end of gesture notification rather than a mouse click.

- Mouse Position
  One of the most important enhancements that should take place in the GBM system, is the mouse position. The problem arose after many user tests, as our study group is not familiar with the drag/drop mouse options. After initiating the gesture, drawing it and finally closing it they couldn't estimate where the mouse position lies. When they wanted to draw a gesture after another, the mouse was lying at the borders of the screen and didn't respond to the shape they were drawing. The solution to this issue is to set the mouse position automatically at the outer most left side of the screen once the user ends a gesture, so as to possess the space to draw another gesture. This solution wasn't easy to implement in Mathematica, because we couldn't find a function to set the mouse position to the new coordinates which we specify. Trying to fix this issue by providing to the user the button "M", this button speaks to the user where the mouse position is and in which direction should the user move the mouse to start a new gesture.

- Enhance speech
  As we put a flash light on the importance of prosody in the mathematical utterance, the speech recognizer of the GBM system was enhanced to speak the head of the functions and to put pauses between different quantities, but that is not enough. It needs more enhancements to reflect the content exactly to the user.

7.2.2 Enhancing the GBM System

This part includes some ideas that could add great use to the GBM system, as this paper is focusing on proving a new concept. The main focus was on the basic issues to test if this concept is applicable to visually impaired people. As this concept is proved, one would think of quality and efficiency in the GBM system as a second step.

- Many Fingers
  The GBM system supports the user interaction with only one finger to draw the gestures. If two or more fingers were to be used the system would get confused. But to apply the collapse/expand functionality as we just noted above, one would preferably need two fingers.

- Integrate with Braille
  To integrate the GBM system with Braille is a marvelous step, because then the user will have all the options and possibilities to interact with the GBM system and other systems simultaneously. As most of the users work with Braille and prefer it this step will allow users to feel at ease. A much greater advantage for integrating the Braille is that users can write/edit to the notebook and not just read it and navigate through ready made contents. They will have the opportunity to write their own notebooks and navigate through them as well.

- Real Touch Slate
  As a basic future step, is to support more advanced touch slates. They have to be bigger than the laptop touch slates for example in the size of the I-Pads. This is important because the user in the current state finds the laptop touch pad so small for interaction and is not familiar with its borders and where it ends. Therefore no doubt a bigger touch pad will be of great help.

- Self Learning Gestures
  One great idea would be to give the user the option to choose their gestures, by defining a set of templates and using them. This way the user would feel more comfortable using gestures which are most applicable to him/her.

- Include Languages
  Including many languages especially in the spoken feedback output to the user, will make the GBM system more international and more users around the world will be able to use it.

- A Separate Program
  On an advanced level one can think that the concept can be generalized to include not only Mathematica notebooks but also Word, LaTeXor PDF documents. And can run as a separate program, with a nice user interface. And can be started from the `Start-up` menu like any other program or editor.

- The Gesture Start Point
  This issue is still a limitation of the gesture recognition part, which is the user has to start the gesture from a definite point and if the user starts it from an opposite side this will be perceived as another gesture that is rotated. Therefore one should give the user more freedom that the gesture is being recognized, independent from where the user starts it or initiates it.

- Speech Input
  As the user is now able to perceive the system output as speech prompts and feedback, in the future one could also think about providing the user with the opportunity to input the mathematical functions and equations by speech as well. This would be the ideal case as the user will interact by the system on speech basis only and maybe need some gestures for small commands. The user will reach the level of human being interaction, which is to interact with the system as interacting with any other human being on speech basis.

This section gave some insights on the possibilities and directions that one can proceed in, as research fields or application fields. So, that one day visually impaired people will find math interactions one of the simplest things.

Chapter 8

Conclusion

8.1 The Sum Up

In conclusion the idea of combining touch gestures and audio speech should be well reflected to the readers. As this concept is new in the field of mathematics and especially for visually impaired people, we saw its impact in the evaluation phase. How it offered a nice environment for the blind people to navigate through mathematical notebooks and access mathematics in a different manner than just the Braille display. The pros and cons were discussed and the challenges that appeared during the implementation and testing. These challenges opened many doors to the future, and is considered as a first step to a new era to help blind people access mathematics in a more realistic advanced way.

As we can't ignore the importance of math in the life of each individual, therefore we understand that math should be accessible to each and everyone. The software tools that were developed to help people access math in an easier more advanced ways as Matlab, Mathematica and many more, the add-ons on these programs that were developed for special purposes as an interface for Braille display and others reflected the huge effort people and technology are consuming to make math easier and more accessible.

On the other hand as we are focusing or directing our effort to help visually impaired people, we noticed that there was a great effort already achieved in this field or direction. We saw the forward steps in technology to help blind people as the Braille display and how it made blind people access the computer and surf the Internet as any other sighted person. It opens the globe to the blind people and enlightens their lives. We further saw the different programs that blind people can interact with, which formats are accessible by them and finally the effort made to publish the articles and papers in accessible formats or the tools that were invented just to convert documents to be accessible to blind people.

Talking about gestures and how they are of big benefit to blind people will open a door which was closed. As this will bring up many applications and fresh ideas to help blind people interact with tools and equipments and achieve more success. Any form of gesture can be of great use as long as we set the boundaries to the visually impaired people, touch gestures as we have already put a flash light on them in this content. Also gestures in the air which can be captured using Kinect camera is a new idea which should be shacked.

In this thesis we had a look on the algorithms which support the gesture recognition, as the $1 algorithm which we based our recognition phase on it. We also had a look on the enhancements introduced to this algorithm, how we added the line recognition and its different directions supporting the horizontal, vertical and diagonal lines by just estimating the angle of this line.

The user manual part was to give the users an idea how to use the GBM system through examples. Figures and tables were provided to reflect to the user the functionalities of this system to navigate and read the notebooks and cells from their head to their leafs.

In the evaluation section we introduced the testing phase of the GBM system. What challenges did the users meet, the difficulty difference of the system to the sighted users than the blind ones. How we managed to add some solutions to aid the navigation of the blind users. For example when the practice mode was added to train the users on the gestures and to know their functionalities. Also the mouse position tracker, because the user can get lost at any time during navigation and not knowing where the mouse position ended. This functionality acts as the light in a dark cave were it estimates the mouse position and advices the user in which direction to move. The full screen option was applied on the interaction notebook which the user draws the gestures on, to assure that the user stays in the correct boundaries of the notebook and not clicking on any other functionality during navigation.

The future work section gave an insight into the future working projects in the mean while and in which direction is the research in this field heading to. This section also covered the enhancements that could be added to the GBM system itself. For example to navigate with two fingers, the collapse/expand command, enhancing the speech of the mathematical functions, including notebooks with styles and combined texts and many more all covered in the future work chapter.

The GBM system was able to prove that the hypothesis question we introduced in the beginning of this thesis is right. And not only right it also proved that combining gestures and speech to help blind users is of great use and efficient. This made it possible to open a door in the future of helping people with visual impairment.

8.2 The Glory of Blind People Capabilities

Not only in science in many other fields, blind people proved their ability to be famous successful and add new ideas.

- Helen Adams Keller (June 27, 1880 - June 1, 1968) an American author, activist and lecturer as she was the first blind person to graduate from college in her time. She also became a famous speaker and author in the world. She is remembered as an activist for people with disabilities.
- Ray Charles (September 23, 1930 - June 10, 2004) was an American pianist and musician who was able to shape the sound of rhythm and blues. He was said to bring a soulful sound to country music and pop standards. The Blues Brothers and Frank Sinatra called him "the only true genius in the business", which is a great justify.
- Marla Runyan (January 4, 1969 - Present) is a marathon runner who is known as being legally blind. She gained her fame as a three-time national champion in the women's 5,000 meters competition.
- Esref Armagan (Born 1953 - Present) a blind painter. He never received any formal schooling or training, but he didn't give up he taught himself to write and print.
- Mr. Armagan is an important figure in the history of picture-making till the day we are living. He was the first to demonstrate that a blind person can develop clear skills equaling the sighted person.

These were a few examples of famous blind people who didn't give up their talents and dreams for the sake of sight. This makes the technology eager to serve them and offer such great people the greatest tools and equipments, so that they can do more.[36]

As sighted people with computer knowledge and new idea developers we should burden ourselves and take the responsibility to offer visually impaired people great tools, systems and applications.

Appendix

Appendix A

Implementation Code

This part includes all the gesture recognition implementation of the $1 Algorithm code and the enhancements in the GBM code, it is implemented in the Mathematica Language.

Resample Function:
```
Resample[p_List,n_Integer]:=
    Module[{points=p,I,D=0,newPoints,d,newP,i=2},
      I=PathLength[points]/(n-1); newPoints=Take[points,1];
     While[i <= Length[points],
         d=Distance[points[[i-1]],points[[i]]];
         If[ D+d >= I,
         newP=LinInter[points[[i-1]],points[[i]],(I-D)/d];
         AppendTo[newPoints,newP];
         points=Insert[points,newP,i];
         D=0,
         D+=d];
         i++];
           If[Length[newPoints]==(n-1),
           Print["Rounding-Error"] ;
          AppendTo[newPoints,p[[-1]]]
            ];
      newPoints]
```

Rotation Function:
```
Centroid[points_List]:=Mean[points]
    IndicativeAngle[points_List]:=Module[{c=Centroid[points]},
      Apply[ArcTan,c-points[[1]]]]
```

Translation Function:
```
    TranslateTo[points_List,pt_]:=Module[{c=Centroid[points]},
      Map[pt-c+#&,points]]
```

Scale Function:
```
BoundingBox[points_List]:=Module[{X,Y},
      {X,Y}=Transpose[points];
      {Max[X]-Min[X],Max[Y]-Min[Y]}]

    ScaleTo[points_List,SquareSize_Integer] :=Module[{B},
```

```
  B=BoundingBox[points];
  Map[#*SquareSize/B&,points]]
```

Optimal Angle and Best Score Function:

```
DistanceAtBestAngle[points_,T_,da_,db_,td_]:=
  Module[{ta=da,tb=db,x1,f1,x2,f2,phi=1/2(-1+Sqrt[5])},
    x1=phi*ta+(1-phi)*tb;
    f1=DistanceAtAngle[points,T,x1];
    x2=(1-phi)*ta+phi*tb;
    f2=DistanceAtAngle[points,T,x2];
    While[Abs[tb-ta]> td,
     If[ f1<f2,
      tb=x2;
      x2=x1;
      f2=f1;
      x1=phi*ta+(1-phi)*tb;
      f1=DistanceAtAngle[points,T,x1],
      ta=x1;
      x1=x2;
      f1=f2;
      x2=(1-phi)*ta+phi*tb;
      f2=DistanceAtAngle[points,T,x2]
        ]
      ];
    Min[f1,f2]]
```

Start Function:

```
StartNB[]:=Module[{Path},
  Path=SystemDialogInput["FileOpen",".nb"];
  NBInteraction=NotebookOpen[Path];
  NB=NotebookGet[NBInteraction];
  SetSelectedNotebook[GestureNB];
  PosNB={1};
  $atNB=True;
   ]
```

Pattern Matching Cases: [1]

```
{__Cell},
l = Length[exp];
Switch[Input,
 6, PosNB = Append[PosNB, 1],
 5, PosNB = Append[PosNB, l],
 8, Speak["A NoteBook of " <> ToString[l] <> " Cells"],
 1 | 2 | 3 | 7, Speak["Invalid move"]; Return[Null]
 ]
,
Cell[CellGroupData[_List, status_]],
```

[1] Mapping Gestures to Numbers: 1=Left Square Bracket, 2=Right Square Bracket, 3=Caret Shape, 4=V Shape, 5=Diagonal Right Line, 6=Diagonal Left Line, 7=Vertical Line, 8=Horizontal Line.

```
Switch[Input,
 6, PosNB = Join[PosNB, {1, 1, 1}],
 5, PosNB =
  Join[PosNB, {1, 1, Length[Extract[NB, Join[PosNB, {1, 1}]]]}],
 2, PosNB = ReplacePart[PosNB, -1 -> PosNB[[-1]] + 1],
 1, PosNB = ReplacePart[PosNB, -1 -> PosNB[[-1]] - 1],
 7, If[Length[PosNB] > 3, PosNB = Drop[PosNB, -3],
  Speak["Invalid move"]],
 8, Speak[
  "Cell Group of  " <>
   ToString[Length[Extract[NB, Join[PosNB, {1, 1}]]]] <> " Cells"],
 3, Speak["Invalid move"]; Return[Null]
 ]
,
Cell[_, _String, ___],
type = Extract[NB, PosNB][[2]];
Switch[Input,
 2, PosNB = ReplacePart[PosNB, -1 -> PosNB[[-1]] + 1],
 1, PosNB = ReplacePart[PosNB, -1 -> PosNB[[-1]] - 1],
 7, If[Length[PosNB] > 3, PosNB = Drop[PosNB, -3],
  Speak["Invalid move"]],
 8, If[type === "Input" || type === "Output",
  Speak["Cell of type  " <> type],
  Speak[Extract[NB, Append[PosNB, 1]]]],
 3, If[type === "Input" || type === "Output", $atNB = False;
  Speak["Cell Level" <> "<silence msec=\"500\"/>"]; PosCell = {};
  CellExp = ToExpression[Extract[NB, Append[PosNB, 1]]],
  Speak[Extract[NB, Append[PosNB, 1]]]; Return[Null]],
 6 | 5, Speak["Invalid move"]; Return[Null]
 ],
```

The Line Template Functions:

```
Points4Line[pX_, pY_, k_, d_] :=
{-((d k - pX - k pY)/(1 + k^2)), -((-d - k pX - k^2 pY)/(1 + k^2))}

FitShapeInto[points_] := Module[{x, L, K, D},
  L = Fit[points, {1, x}, x];
  D = L[[1]];
  K = L[[2]]/x;
  {K, D}
  ]

LineTemplate[points_] :=
 Module[{K, D, FirstPoint, FirstX, FirstY, e, LastX, LastY, StartLine,
    EndLine},
  {K, D} = FitShapeInto[points];(*Gives equation of a line *)
  FirstPoint = First[points];
  FirstX = FirstPoint[[1]];
```

```
  FirstY = FirstPoint[[2]];
  e = Length[points];(*Gives Size*)
  LastX = points[[e, 1]];
  LastY = points[[e, 2]];
  StartLine = Points4Line[FirstX, FirstY, K, D];
  EndLine = Points4Line[LastX, LastY, K, D];
  {StartLine, EndLine}
  ]


(*Construct a line between 2 points*)
MyLine[{X_, Y_}, Val_] := Module[{step, i},
  step = (Y - X)/(Val - 1);
  Table[X + i*step, {i, 0, Val - 1}]
  ]


LinePathDistance[points_, T_] := Module[{i},
  Sum[Distance[points[[i]], T[[i]]], {i, Length[points]}]/
   Length[points]]
```

Appendix B

Lists

List of Abbreviations:

GBM	Gesture Browsing of Mathematics
GSS	Golden Section Search
MAWEN	Mathematical Working Environment
RTF	Rich Text Format
ASTER	Audio System for Technical Reading
AFL	Audio Formatting Language
MSAA	Ms Active Accessibility

List of Figures

1.1 Mathematical formula displayed in 2D 2
1.2 Mathematical formula displayed in linear form 3
1.3 The Braille Display 4
1.4 Visually impaired users using the Braille display 4

3.1 Project Steps 16
3.2 Old Projects Structure 17
3.3 Current Projects Structure 18

4.1 Notebook X, Mathematica Notebook Example 21
4.2 Tree Structure of Notebook X 22

5.1 Laptop Touch-Pad 26
5.2 The shapes used in the project 29
5.3 Right square bracket gesture 30
5.4 Square bracket shape without Resampling 31
5.5 Resampling the right square bracket with $10, 30, 60$ points respectively 31
5.6 Centroid of the right square bracket 32
5.7 Rotation right square bracket gesture 32
5.8 Scaling right square bracket with scale 100,200 and 500 respectively 33
5.9 Right square bracket before translation and after translation 33
5.10 Golden Ratio: $a + b$ is to a as a is to b 34
5.11 Right square bracket in orientation A 35
5.12 Right square bracket in orientation A resulted recognition 35
5.13 Right square bracket in orientation B 35
5.14 Right square bracket in orientation B resulted recognition 36
5.15 Vertical Line 36
5.16 Vertical Line resulted in Caret recognition 37
5.17 Horizontal Line 37
5.18 Vertical line resulted in Caret recognition 37
5.19 V Shape different start point 38
5.20 V shape perceived as a Caret shape 38
5.21 Right square bracket perceived as left square bracket 39
5.22 The type of lines defined, diagonal left, diagonal right, vertical and horizontal lines 40
5.23 Range of Angles 41

6.1 User testing the laptop touch-pad 48

6.2 User testing the gestures and receiving sound feedback 49
6.3 Mouse Position Possibilities 50

Bibliography

[1] Britannica. Britannica concise encyclopedia. http://www.britannica.com/, May 2013.

[2] Laure Gouba. The importance of mathematics in everyday life. PDF Presentation, 2008.

[3] Annenberg Foundation. Math in daily life. http://www.learner.org/interactives/dailymath/, May 2013.

[4] Dominique Archambault. Non visual access to mathematical contents: State of the art and prospective. In *Proceedings of the WEIMS Conference*, pages 43–52, 2009.

[5] Dominique Archambault, Bernhard Stöger, Donal Fitzpatrick, Klaus Miesenberger, et al. Access to scientific content by visually impaired people. *Digital journal of CEPIS (Spanish version was published in Novática)*, 8(2):29–42, 2007.

[6] Carolyn Ruth Northcott. *The impact of braille contractions on the quality of diagnostic information gathered using informal reading inventories with beginning braille readers*. PhD thesis, University of British Columbia, 2002.

[7] Jasonpearce. Braille book with shadows and texture. http://images.cdn.fotopedia.com/flickr-136206296-original.jpg, Apr 27 2006. Braille Display.

[8] Bernhard Stöger. Personal communication, 2013.

[9] Dominique Archambault, Bernhard Stöger, Mario Batui, Claudia Fahrengruber, and Klaus Miesenberger. Mathematical working environments for the blind: what is needed now? *Interaction et usages des modalités non visuelles, accessibilité des contenus complexes*, page 36, 2007.

[10] Cristian Bernareggi and Dominique Archambault. Mathematics on the web: emerging opportunities for visually impaired people. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, pages 108–111. ACM, 2007.

[11] Donal Fitzpatrick. Mathematics: how and what to speak. In *Computers Helping People with Special Needs*, pages 1199–1206. Springer, 2006.

[12] Douglas J Gillan, Paula Barraza, Arthur I Karshmer, and Skye Pazuchanics. Cognitive analysis of equation reading: Application to the development of the math genie. In *Computers Helping People with Special Needs*, pages 630–637. Springer, 2004.

[13] Carole Beal and Erin Shaw. Working memory and math problem solving by blind middle and high school students: Implications for universal access. In *Society for Information*

Technology & Teacher Education International Conference, volume 2008, pages 5011–5016, 2008.

[14] Bruno Buchberger. International conference on computers helping people with special needs. In *Mathematics is Blind*, 2012. Personal Communication.

[15] Bruno Buchberger. Personal communication, 2013.

[16] Arthur I Karshmer, Chris Bledsoe, and Paul Stanley. The architecture of a comprehensive equation browser for the print impaired. In *Computers Helping People with Special Needs*, pages 614–619. Springer, 2004.

[17] Wolfram Mathematica. Wolfram mathematica documntation center. http://reference.wolfram.com/mathematica/guide/Mathematica.html.

[18] Klaus Miesenberger. Personal communication, 2013.

[19] Reeta Singh. Blind handicapped vs.technology:how do blind people use computers? *International Journal of Scientific & Engineering Research*, 3, 2012.

[20] MatlabWorks. Matlab the language of technical computing. http://www.mathworks.de/products/matlab/.

[21] Wolfram Mathematica. What is mathematica ? http://www.wolfram.com/mathematica/.

[22] R Foundation. R-project. http://www.r-project.org/.

[23] Indian TEX Users Group. Latex tutorials a primer. http://www.tug.org.in/tutorials.html, Floor III, SJP Buildings, Cotton Hills Trivandrum 695014, India, 2003.

[24] William Park. Latex2e package for typesetting braille. ftp://ftp.tex.ac.uk/ctan/pub/tex/macros/latex/contrib/braille/braille.html#Sec1, April 1999.

[25] A Jonathan R Godfrey. Putting it all togethera blind persons perspective on developing a toolbox for document preparation. *Inftyreader. org*, 2012.

[26] Wolfram Mathematica. Blind optical physicist pursues technical career with the aid of mathematica. http://www.wolfram.com/mathematica/customer-stories/blind-optical-physicist-pursues-technical-career-with-the-aid-of-mathematica.html.

[27] Alysha Jeans Al Maneki. Latex: What is it and why do we need it? *Future Reflections*, Volume 31(2), 2012.

[28] David M Thompson. Latex2tri: Physics and mathematics for the blind or visually impaired. In *Proceedings from the 20th Conference on Technology and Persons with Disabilities*. Citeseer, 2005.

[29] Bernhard Stöger, Klaus Miesenberger, Mario Batusic, and Dominique Archambault. Multimodal interaction with mathematical formulae: Access, communication and support for blind people in doing mathematics. *Interaction et usages des modalités non visuelles, accessibilité des contenus complexes*, page 48, 2005.

[30] Katsuhito Yamaguchi and Masakazu Suzuki. Accessible authoring tool for daisy ranging from mathematics to others. In *Computers Helping People with Special Needs*, pages 130–137. Springer, 2012.

[31] Jacob O Wobbrock, Andrew D Wilson, and Yang Li. Gestures without libraries, toolkits or training: a $ 1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 159–168. ACM, 2007.

[32] Hartley Hyde. The golden ratio. *Australian Mathematics Teacher*, 60(1):30–31, 2004.

[33] Wolfgang Windsteiger. Personal communication, 2013.

[34] Helder Ferreira and Diamantino Freitas. Audiomath: using mathml for speaking mathematics. *XATA05*, 2005.

[35] Sieb Nooteboom. The prosody of speech: melody and rhythm. *The handbook of phonetic sciences*, (5):640–673, 1997.

[36] Disabled World. Famous people with vision impairments past and present. *Disabled World - Disability News and Information*, 2008.